ENGLISH PREMIER LEAGUE SCHEDULING USING SIMULATED ANNEALING

Daniele Alexandre Ferreira

A research report submitted to the Faculty of Engineering and the Built Environment, of the University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg 2014

University of the Witwatersrand, Johannesburg

School of Mechanical, Industrial and Aeronautical Engineering

DECLARATION

Name:	**Daniele A.C. Ferreira**	Student no:	**388882**
Course no:	**MECN 7018**	Course Name:	**Research Report**
Submission Date:	**15 October 2014**	Project Title:	**English Premier League scheduling using Simulated Annealing**

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong;
- I confirm that the work submitted herewith for assessment in the above course is my own unaided work except where the I have explicitly indicated otherwise;
- This task has not been submitted before, either individually or jointly, for any course requirement, examination or degree at this or any other tertiary education institution;
- I have followed the required conventions in referencing the thoughts and ideas of others;
- I understand that the University of the Witwatersrand may take disciplinary action against me if it can be shown that this task is not my own unaided work or that I have failed to acknowledge the sources of the ideas or words in my writing in this task.

Signature: Date: 15 October 2014

ABSTRACT

This is the first known attempt at scheduling the English Premier League (EPL), which is a NP-hard problem, in the literature. In this research an initial schedule is created using a 'polygon' construction method, a method which originates in graph theory. Two distinct simulated annealing metaheuristic solving methodologies are then created to improve this initial schedule. One method is based on a temperature schedule, finite epoch length and reheats while the other is based on a gradually reducing temperature schedule and non-finite epoch length. These two methods were evaluated with respect to solution quality (total penalty), reliability (variation of solution quality over numerous trials) and speed. The official schedule used by the EPL organisers was used for comparison. It was found that the first method produced comparable results, while the second produced improved results. The second method was validated over three seasons and consistently performed well. The findings in this research can be used as the maiden real-world framework and benchmark for the unsolved EPL scheduling problem in the sports scheduling literature.

CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background and Motivation

Each year the EPL association has the challenging task of scheduling the EPL, the football league with the largest broadcasting value and following in the world [1], for the upcoming season. It is a process which begins at the start of each year by ruling out international, European and domestic cup dates [2]. Schedulers must then compile a schedule which balances the requirements, often conflicting, of various stakeholders which include the clubs, police, fans and league owners [2].

A favourable schedule may provide teams with the best chance to compete in all lucrative competitions. Conversely, a weak schedule would hamper their chances of progressing in lucrative cup competitions whilst maintaining a good league position. For example, two highly-rated clubs may be unfortunate to be scheduled against one another prior to a highly competitive and lucrative European knock-out game. To maintain league position, these teams will compete in the exhausting league game which will compromise their fitness levels for the following, equally challenging, European cup game.

As a result, it is in the best interest of both the clubs (to win lucrative competitions) and the EPL association (to promote the EPL) to compile a favourable and attractive schedule. While sports scheduling literature does exist, there is no such research attempting to schedule the EPL, which has a scheduling formulation substantially different to other football leagues [3], [4]. A formulation and maiden benchmark for the EPL instance needs to be established.

1.2 Research Scope

In England, the top four football leagues are scheduled together. The scope of this research will focus on the isolated case of scheduling only the EPL into slots (match days). It is assumed that the league and broadcasters then assign dates and kick-off times for these slots. Consistent with the Premier League, the schedule will be influenced by domestic and European cup competitions. Also, this research considers only those events which could be found in the literature such as traditional festivals. It does not consider other events which may affect a fixture such as a concert in close proximity to a stadium or logistical concerns of fans using the same transport routes. Not only is it difficult to gather all such data, the effects of these adverse events can be mitigated by shifting fixtures to either the Saturday or Sunday.

2 OBJECTIVES

The primary objectives of this research are to:

- Identify EPL scheduling stakeholders and their respective requirements
- Develop an EPL scheduling technique
- Validate the EPL scheduling technique using three instances

3 LITERATURE SURVEY

The literature survey is aimed at gaining insight into three aspects. Each is necessary in order to understand the nature of the EPL scheduling problem and to identify the best approach to formulate and solve it.

First, the nature of the EPL needs to be understood. This entails the identification of the EPL's constituents and their inter-relationship. Second, the methodology currently used by the EPL should, if possible, be identified. This allows a researcher to compare methodologies and scheduling techniques. Lastly, sports scheduling literature is researched to identify: the types of scheduling problems faced by sports schedulers, the different scheduling techniques employed by researchers, the published benchmarks and the computational complexity of sports scheduling problems.

3.1 Nature of the EPL

The nature of the EPL is concerned with identifying the league structure, the different tournaments involved and which teams qualify for the different tournaments. Also of relevance are the potential earnings for each tournament. This way one can understand the requirements of the clubs and league owners as well as the relative importance of each.

3.1.1 Tournaments and composition

- **Domestic League**

The EPL consists of 20 teams. Each team participates in a double round-robin format with both home and away legs between any two teams. A total of 380 matches are to be scheduled in a season with each team playing 38 matches[1].

In the final league standing of each season, the bottom three placed teams are relegated to the lower Championship division. Three teams in turn are promoted via direct qualification and promotional play-off qualification games from the Championship division.

- **Domestic Cups**

The Football Association Cup (FAC) is a knockout competition with six preliminary rounds and eight final rounds called 'proper' rounds. EPL teams, however, only enter in the third proper round [5]. Fixtures are replayed if a game ends in a draw. Extra time and penalties result if the replay does not yield a winner.

[1] 20 teams play 38 matches each with two teams participating in each match equates to 20x38/2 = 380 matches.

The Football League Cup (FLC) features seven knockout rounds. Round six (the semi-final) however, features a home and an away leg [6]. Teams competing in the European cups enter in round three while the rest of the EPL teams enter in round two [6].

The FAC consists of all teams which are part of the English Football Association (FA). However, for lower ranked teams in the association, places are subject to availability and ground conformance [7]. Participation numbers in the competition vary each season with 737 teams entering the 2013/14 competition [8]. On the other hand, the FLC competition only features teams from the top four divisions, totalling 92 teams [6].

- **European Cups**

The premier Union of European Football Associations (UEFA) Champions League (UCL) follows a similar format to that of the 32 team Fédération Internationale de Football Association (FIFA) World Cup, except that each matchup, excluding the final, consists of a home leg and an away leg. The top two teams from each group, consisting of four teams, progress to the knockout rounds. The teams qualifying third in each group proceed to the less lucrative UEFA Europa League (UEL).

Like the UCL, the UEL follows a two-leg format, except for the final. It begins with a group stage consisting of twelve groups [9]. The top two teams of each group proceed to the knockout rounds, starting with the round of 32, where the balance of eight teams are transferred from the UCL [9].

Qualification requirements for European Cups are depicted in Table 1. The requirements are based on the previous season's results.

Table 1: European Cup qualification requirements for EPL teams [9], [10]

Competition	Requirement	Qualification type
UCL	EPL position 1 – 3	Automatically qualify for UCL
	EPL position 4	Enter UCL preliminary play-off round
	UCL winner	Automatically qualify for UCL
UEL	EPL position 5	Enter UEL preliminary play-off round
	FAC winner	Automatically qualify for UEL
	FLC winner	Enter UEL preliminary third round
	UEL winner	Automatically qualify for UEL

Various contingencies exist which affects who and how many teams may qualify for the European Cups. The first is the possibility that a team wins the UCL but finishes the EPL season in fifth position. In this case, this team will qualify for the upcoming UCL as the title holder with the team which placed fourth in the EPL taking the now vacant UEL position [10]. Alternatively, if the UCL title holder finishes the league outside both the UCL and UEL positions, this team will again qualify for the UCL as winners and an additional UEL place will be granted to the league for the fourth placed EPL team [10]. With regards to the UEL, should the FAC winners qualify for the UCL then the FAC losing finalists will qualify for the UEL [10]. Should the FAC winners (or losing finalists if applicable) and the FLC winners already have qualified for either the UCL or UEL through league position, the next best placed teams in the EPL will qualify [10].

- **European and World honours**

The UEFA Super Cup is a single fixture played between the current UCL and UEL cup holders [11]. UCL winners also compete in the Club World Cup tournament which comprises all continental cup winners along with the host nation's league champions [12]. Continental cups include the European, African, North American, South American, Oceania and Asian champion's league [12]. UCL champions enter the Club World Cup directly at the semi-final stage [12].

3.1.2 Potential earnings

- **Domestic League broadcasting earnings**

Clubs receive money from sponsorships, gate revenue and broadcasting revenue. Broadcasting revenue contributes largely towards a club's earnings. While the international broadcasting revenue stream is divided equally amongst the EPL teams, the local broadcasting revenue is divided on a 50:25:25 split [10]. This means that 50% of this revenue stream is equally divided between the 20 EPL clubs, 25% is distributed according to a club's final league position and the remaining 25% is distributed as a facilities fee which is based on the number of matches shown on television involving a particular club [10].

Negotiations for domestic television rights are competitively bid for in a three year cycle. In the 2010-2013 cycle, the value was £1.8bn [13] whereas the bid for the 2013-2016 cycle rose to approximately £3.4bn [1]. On the other hand, overseas television value, which covers over 200 countries, has increased by over 50% to £2.2bn for the new cycle [1]. Combined, the domestic and overseas broadcasting deal is valued at around £5.5bn for the 2013-2016 cycle. Based on this revenue distribution mechanism, EPL clubs will receive between £60m and £90m each year in the 2013-2016 cycle [1].

To place the EPL's value in context, the total revenue for the top five European leagues in the 2011/12 season were: England €2.9bn, Germany €1.9bn, Spain €1.8bn, Italy €1.6bn and France €1.1bn [1].

- **Domestic Cup earnings**

Table 2 presents the distribution of the FAC prize money.

Table 2: FAC prize money 2012/13 season [14]

Round	Prize money
Third Round Proper winners	£ 67.5k
Fourth Round Proper winners	£ 90.0k
Fifth Round Proper winners	£ 180k
Sixth Round Proper winners	£ 360k
Semi-Final losers	£ 450k
Semi-Final winners	£ 900k
Final runners-up	£ 900k
Final winners	£ 1.80m

The total prize money distributed for the 2010/11 FAC [15] was £15m while the total broadcasting rights were £9.6m for a grand total of around £25m. The winning prize money is relatively small but money is accumulated as a team progresses. The total accumulated winnings received by Manchester City who won the 2010/11 FAC was £3.4m [15]. An additional £0.9m was received in TV payments by the club with gate revenue not known. To get an idea of the gate revenue, Stoke City received over £1.1m in gate revenue for reaching round six in the same season [15].

There is little money in the FLC cup in terms of prize money. Unlike the FAC where prize money is distributed depending on how far a team progress, the FLC prize money is distributed between the top four teams. Winners receive £100k, losing finalists receive £50k and losing semi-finalists each receive £25k [16]. This prize money is relatively low and Conn [16] notes that this prize money is less than a top EPL player's wage.

With regards to additional FLC revenue streams, the finalists each receive £250k from the broadcasting money and 45% each of the gate revenue (10% to facilities and administration costs) [16]. In the 2012/13 FLC, it was estimated that Bradford City had accumulated, through all revenue streams, a total of £1.3m prior to the final and a further £1.0m for participating in the final [16]. As a consequence of the low earnings, EPL teams use the competition to give young and fringe players a chance to play.

- **European Cup earnings**

The UCL is a lucrative competition as can be seen by the earning potential in Table 3 below. Clubs merely participating in the preliminary play-off round received a fixed €2.1m in the 2012/13 season [17]. Money is distributed to qualified teams for appearance, performance in the group stages, progression prize money and TV broadcasting money [18].

Table 3: Prize money per round for the 2012/13 UCL and UEL season [17], [19]

Round	UCL Prize money	UEL Prize money
Round of 32	-	€ 200k
Round of 16	€ 3.5m	€ 350k
Quarter-finals	€ 3.9m	€ 450k
Semi-finals	€ 4.9m	€ 1.0m
Runner-up	€ 6.5m	€ 2.5m
Winners	€ 10.5m	€ 5.0m

Excluding TV revenue, based on performance, clubs can earn between €8m and €12m in the group stage [18]. TV revenue is distributed according to the club nation's TV broadcasting value. Since England has the largest broadcasting value of all European leagues, the English FA receives a large portion of this revenue stream. The FA then distributes 50% of this money based on the participating clubs EPL position in the previous season and the remaining 50% gets distributed based on a clubs progression in the current UCL. TV earnings can range from €1.0m to €30m with €15m received by the lowest earning English team in the 2011/12 season [18].

A total of €755m was distributed to 32 clubs participating in the 2011/12 UCL season [18]. The total money earned by Chelsea FC, winners of that season, was €60m (with approximately €30m from the TV market pool) while the runners-up, FC Bayern München, received €42m [18]. Even though Manchester United did not progress past the group stage, they received €35m, largely as a result of their TV broadcast earnings [18].

Although the UEL is not as lucrative as the UCL, its earning potential is relatively large compared to the domestic cups. Clubs who entered the 2012/13 UEL preliminary qualification rounds received €100k for each round [19]. Like the UCL, clubs receive prize money based on participation, group stage performance, progression and broadcasting money [20]. Clubs can earn between €1.0m and €1.7m for participation and performance bonuses while TV money can range from €100k to €1.5m [20].

A total of €150m was distributed to the 56 clubs participating in the 2011/12 UEL season [20]. The total money earned by Atletico Madrid who won the 2011/12 UEL was €11m while the runner-up, Athletic Club, received €10m [20].

- **European and World honour earnings**

Winners of the 2012 UEFA Super Cup earned €3.0m in prize money while the runners-up received €2.2m [17]. With regards to the Club World Cup, the total prize money distributed in the 2008 Club World Cup was $16.5m with the winners pocketing $5.0m as shown in Table 4 [21]. For the 2012 Club World Cup, the total prize money was also $16.5m [22] and it is therefore assumed that the prize money distribution remained the same. Since the UCL winners only enter the Club World Cup at the semi-final round, they receive a minimum of $2.0m.

Table 4: Prize money distribution for the 2008 Club World Cup [21]

Position	Prize money
Winners	$ 5.0m
Runner-up	$ 4.0m
Third	$ 2.5m
Fourth	$ 2.0m

3.2 Current EPL Scheduling Methodology

It is necessary to identify the current methodology and scheduling technique used by the Premier League to schedule the EPL. This would allow researchers to compare models and results. This section attempts to identify the methodology employed by interpreting information gathered from various sources. It appears that the methodology can be separated into four steps which include eliminating reserved dates, data gathering, creating a draft and finalisation.

- **Step 1 – reserved dates**

The football authorities are concerned with scheduling the EPL and lower divisions which form part of the Football League association [23]. Together they form the top four divisions in England. With the EPL in particular, 380 fixtures are to be scheduled around the international (FIFA), European (UEFA) and domestic cup (FA) games. Hence, the dates for these matches must be known and set aside. The available dates which the EPL can then be assigned to is determined. These dates are usually known at the start of the year, approximately eight months before the start of the new season in mid-August [23].

- **Step 2 – data gathering**

The second step concerns finding the requirements of various stakeholders. Schedulers attempt to identify and gather all the demands of the various stakeholders to ensure that no team, set of supporters, or police are unduly penalised [2].

Constraints affecting the schedule include the so-called ‘Premier League rules’. The ‘golden’ rules identified from the literature are listed:

a. No team should have more than two successive home or away matches. This aims to ensure cash-flow for clubs through regular gate revenue, pitch conservation by avoiding overuse, and less travel for fans [24].

b. All teams should have an alternating sequence of home and away fixtures at both the beginning and end of the season (i.e. for the first and last two games) since it would be unfair for a team to finish with two away fixtures, especially if they are looking for points [2].

c. All teams must have an alternating sequence of home and away fixtures over the Boxing Day and New Year’s Day period [2].

d. Strength fixtures (fixtures between the best teams) and local derbies should be avoided on Boxing Day and New Year’s Day [25]. Since good stadium attendance is guaranteed for these days, derbies and attractive strength fixtures should take place some other time [26].

e. Strength fixtures are to be avoided on the opening day of the season [2].

f. The travelling distance to away fixtures on Boxing Day and New Year’s Day should be minimised wherever possible to lessen the inconvenience to travelling supporters, since public transport is limited during these days [25].

g. Minimise the case where teams from the same geographic area travel on the same public routes in order to avoid potential conflict [25].

h. Each team should have an equal number of home and away Bank Holiday fixtures [2].

i. Matches either side of a FAC fixture should not both be away from home [27].

Furthermore, a questionnaire is sent to all teams and the police around March to request three things [23]. The first question concerns any dates a club or the police wish to avoid a home game [23]. For example, Liverpool do not wish to have a home game during the same weekend as the Grand National horse racing event which happens in close geographic proximity and might have an effect on stadium attendance and revenue. The second question concerns which teams a club wishes to be ‘paired’ with [23]. Many teams, usually local rivals, are ‘paired’ when stadiums are in close proximity (described further in the next paragraph). The third question asks whether there are any teams a club does not wish to play at home on Boxing Day (or police request) [23]. For example, police may request that fixtures notorious for hooliganism not to be scheduled for Boxing Day.

It is not desirable for paired teams like Manchester United and Manchester City, who both reside in Manchester, to both have home fixtures on the same day. The justification is that handling both fixtures would stretch and put excessive strain on local services [2]. Paired teams for the 2013/14 season include: Newcastle and Sunderland, Everton and Liverpool, Manchester United and Manchester City, Aston Villa and West Brom, Swansea City and Cardiff City, Tottenham Hotspurs and Arsenal, and Chelsea and Fulham. Teams not paired include West Ham, Crystal Palace, Stoke City, Hull City, Norwich City and Southampton. Appendix A graphically illustrates the geographic location of each team’s stadium around the UK.

There are around 90 policing restrictions which affect the schedule [2]. Such restrictions include the Oxford-Cambridge Boat race affecting Chelsea and Fulham and the Grand National affecting Liverpool. Other restrictions include logistical restrictions such as too many supporters, or rival supporters, using the same transportation routes [2]. Most of these restrictions are resolved simply by interchanging a Saturday fixture to Sunday (or vice-versa). A special case is however made for the Boxing Day and New Year’s Day fixtures. Due to the limited availability of public transport on these days, these fixtures are manually inserted into the schedule first. Fixtures for the rest of the season are then scheduled around these.

- **Step 3 – draft schedule**

The factors from Step 2 are then processed and fed into an in-house software system developed by Atos who have had the EPL scheduling contract for 30 years [24]. A draft is then output using a methodology adopted in 1982 called ‘sequencing’. Glenn Thompson of Atos, who has been heading the scheduling since 1992, explains that the season is broken down into five parts called ‘sets’ which are reversed in the second half of the season [24]. Each club is then placed on a pairing grid. For each slot, knowing which teams are at home and which are away, the software will determine the fixtures to take place [23].

Thompson explains that once the parameters have been entered into the software, draft schedules can be generated in as little as 10 seconds but can take up to 30 minutes using a basic laptop [24]. However, this was not always the case. Thompson says that in the past the staff would let the software run overnight or even until the next afternoon for a schedule. The software was originally built by the predecessor to Atos, called CAP, and ran on the first generation of IBM Basic PC’s [24]. Given the limited processing power available in the 1980s, this could be a slow process. In recent times, Atos has migrated its scheduling software to Microsoft Visual Basic.NET [24].

- **Step 4 – review and finalisation**

Representatives from the Premier League, Football League and Atos review the draft schedule. This review ensures that wherever possible, all stakeholder requirements have been met. If there are any concerns, the software is capable of finding alternative dates for fixtures to be moved to. However, one change can lead to 40 other changes to the schedule. Therefore, sometimes a new schedule is generated. According to Thompson [23], approximately 85% of requests are accommodated.

A meeting with the Fixtures Working Party then takes place. Representatives include the FA, Premier League, Football League, Football Supporters’ Federation and Atos. The Fixtures Working Party pay particular attention to fan concerns such as travel on midweek fixtures and the Boxing Day and New Year’s Day fixtures, while also looking out for potential issues regarding opening and closing fixtures of the season [26]. This schedule is then considered by various policing authorities which includes the Association of Chief Police Officers, National Criminal Intelligence Service and British Transport Police [2].

The review process can take several days to complete [24]. Once all reviews and adjustments have taken place, Atos’s work is finished and the provisional schedule is published (mid-June). Although the fixtures between two teams cannot be changed after this, match dates and times are subject to change. League owners then take responsibility for postponing and rescheduling fixtures caused by bad weather or cup replays [24].

3.3 Theoretical Research

3.3.1 Available literature

An annotated bibliography by Kendall et al. [28] is an excellent starting point for researchers interested in sports scheduling. It references over 160 papers with a paper published in 1968 being the earliest considered. In the article, it notes that the interest in this field is increasing with around fifteen papers being published each year between 2005 and 2009. The annotated bibliography consolidates the papers referenced and highlights the types of problems and scheduling techniques used to solve them. It also discusses different applications in the sports scheduling world. A similar structure is used in this section (Section 3.3) with the application focus on football league scheduling.

Key features of sports scheduling range from optimising revenue to optimising logistics and fairness. With multiple stakeholders there can be conflicting needs, like broadcasting revenue versus logistic optimisation. Schedulers therefore have to obtain constraints and objectives of all stakeholders and balance trade-offs in optimising schedules. A well-balanced and high quality schedule may even make a league more attractive. For example, spreading attractive games throughout the season can increase TV viewership which is an objective for TV broadcasters in the Belgium league [3].

Sports scheduling problems have attracted interest from multidisciplinary fields which includes operations research, constraint programming, graph theory, combinatorial optimisation, and applied mathematics [28]. Readers may find a library published by Sigrid Knust [29] useful, called the “Classification of literature on sports scheduling” library. This library classifies an up to date list of sports scheduling references.

3.3.2 Terminology

To be consistent with the work completed previously in the field of sport scheduling, the terminology used in this research will conform to that used in “Round-robin scheduling - a survey” by Rasmussen and Trick [30]. This article noted that the terminology used in the literature up to 2007 was highly inconsistent and therefore presents a unified terminology presented next:

Round-robin (RR): A round-robin tournament is a tournament where each team meets every other team a fixed number of times (a 1RR refers to a single round-robin).

Double round-robin (2RR): A round-robin tournament where every pair of opponents meet twice. It is often required that the two games between every pair of opponents occurs at each other's venue.

Slots: When scheduling a tournament, the games must be allocated to a number of timeslots, called 'slots' (i.e. round of matches), in such a way that each team plays at most one game in each slot.

Compact or relaxed schedule: The tournament is said to be compact when (*n-1)* slots are required to schedule a 1RR for an even number of teams *n* while it is relaxed when more slots are available.

Timetable set (T Set): The allocation of games to slots. Each row *t* of the timetable corresponds to a team while the column *s* corresponds to a slot. Each entry is the opposition of team *t* in slot *s*.

Venue, home games and away games: Teams often have a home venue. A team plays home games at their home venue and away games at an opponent's venue.

Balanced schedule: For a 1RR, the difference between the number of home games and away games played by each team is no more than one. A 2RR is balanced if it is required that two games between every pair of opponents occur in opposite venues.

Home and away pattern set (HAP Set): The sequence of home and away games according to which a team plays during the tournament. Each entry in the *HAP Set* contains either an *H* or an *A* (alternatively *1* or *0*).

Break: A tournament with an alternating pattern of *H* and *A* games is considered attractive and a break from this corresponds to two consecutive slots of home (or away) games.

Trip or home stand: A sequence of consecutive away games is called a 'trip' while a sequence of consecutive home games is called a 'home-stand'.

Tour: An entire row of the schedule defines a tour for the corresponding team.

Complementary patterns: Two row vectors in a *HAP Set* are said to be complementary if the first pattern has an away game when the second pattern has a home game and vice versa.

Equitable pattern set: If all teams have an equal number of breaks.

Feasible pattern set: A *HAP Set* for which a corresponding *T Set* exists.

Schedule: The superimposition of a *HAP Set* and a corresponding *T Set* constitutes a schedule.

Mirrored schedule: When the first and the second half round-robins are identical except that the home venues are reversed.

Irreducible schedule: A schedule for a 1RR is irreducible when at most one opponent in each game has a break.

The following are graph theory definitions which are used in some sports scheduling research:

Graph (G): A graph $G = (V,E)$ where V is a finite set of vertices and E is the set of edges in G.

A matching in G: A set of independent edges (non-adjacent edges).

A perfect matching in G: A matching in which all the vertices in V are incident to an edge.

One-Factor graph: The graph induced by a perfect matching is a one-regular graph (for all nodes the degree is, $deg = 1$), called a 'one-factor' graph.

One-Factorisation: The partitioning of a graph G into one-factor graphs.

3.3.3 Sports scheduling problems

Kendall et al. [28] defines the basic sports scheduling problem as follows. A league consisting of an even number of teams n, requires each team to play against every other team a set number of times l (l = 2 for a 2RR). The number of slots available to schedule these $n(n-1)l/2$ fixtures is equal to $(n-1)l$ for a compact schedule. Often in a 2RR, the season is partitioned into two halves of 1RR's where each fixture must occur once in each half but with reverse venues. A sports league schedule usually consists of a timetable *T Set* representing the opponent schedule and a home-away pattern *HAP Set* representing the venues. Therefore, for each slot $s = 1,...,(n-1)l$, a scheduler must assign the teams i, $j \in \{1,...,n\}$ to play one another in the *T Set* as well as the venues to be played at in the *HAP Set*.

Schedulers apply various scheduling techniques to the different sports scheduling problems. The most common types of scheduling problems are described next.

- **Graph theory**

Early research into sports league scheduling was based on graph theory [28]. A complete graph K_n can represent a 1RR tournament [31]. In the complete graph K_n, the set of vertices $V = \{1,2,\ldots,n\}$ represents the teams and the set of edges $E = \{ij,\ldots\}$ represents the fixtures between teams $i, j \in \{1,...,n\}$.

Edge colouring with $(n-1)$ colours for each incident edge of all vertices in a complete graph partitions the edges into one-factors consisting of $n/2$ non-adjacent edges [28]. Each one-factor corresponds to the fixtures of each slot. Venues can be distinguished by adding arcs (i,j), thereby creating a digraph. A fixture in a digraph is held at the home venue of team i.

Graph theory can be useful in that certain properties, which can be proved mathematically, are established. De Werra [32] provides theories and proofs for sports scheduling using graph theory. One such property is that a 1RR schedule with an even number of teams 2n has at least $(2n-2)$ breaks. For example, a scheduler concerned with break minimisation of 20 teams would know that a 1RR schedule is optimal if the number of breaks is $20-2 = 18$. Some graph-theoretical models for scheduling sports leagues concerning the break minimisation problem (discussed later) are given.

- **Latin Squares**

A 1RR *T Set* may be represented by a $n \times (n-1)$ matrix. However, if the T Set is made square by adding a column vector with the teams $1,..., n$, it can form a latin square structure [28]. A latin square is a structure such that no number occurs more than once in any column or row [28]. Latin squares are also symmetrical, making it ideal for round-robin scheduling. This means that if team i plays against team j in slot s then team j plays against team i in the same slot (i.e. considering timetable set T, $T_{is} = j$ if and only if $T_{js} = i$) [28].

Dinitz [33] discusses the use of Latin squares in solving his mother-in-laws golf scheduling problem which he subsequently called the Joyce Cook golf league problem. Broadly, this problem entails the scheduling of 39 golfers to play in threesomes over ten weeks and with 13 different starting positions. No golfers are to start in the same starting position twice and golfers must play with different partners each week.

- **Model development**

Generally, sports scheduling problems consist of a set of constraints and aim to achieve some objective [28]. Problems found in the literature are either based on real-world applications or have been introduced artificially for researchers.

Some real-world problems found in the literature may appear similar but are in fact different with tailored constraints and objectives [28]. For example, the German and Belgium leagues appear to be similar in that each league consist of 18 teams in a mirrored 2RR but the soft and hard constraints differ as outlined in Nurmi et al. [3]. This makes it difficult to provide a standard objective and set of constraints.

Kendall et al. [28] notes that some sports scheduling problems, which do not focus on real-world applications, have been introduced for the benefit of research. Working with a common artificial model, researchers can compare results for different instances using a benchmark (to be discussed in Section 3.4). A prime case where problems have been introduced for research is the travelling tournament problem.

- **Travelling Tournament Problem (TTP)**

The TTP problem aims to minimise the total distance travelled by all teams in a tournament. Easton et al. [34] introduced the problem to minimise the distance travelled by Major League Baseball teams in the United States. The TTP is based on road trips where teams can play multiple matches before returning home as a result of the large geographical spread of teams. Kendall et al. [28] state that usually, constraints restrict the minimum and maximum number of consecutive games that can be played at home or away (i.e. trip and home-stand limits). Also, constraints prohibit ‘repeaters’ which disallows teams from facing each other in consecutive slots [28].

- **Break minimisation**

One of the most researched sports scheduling problems is the break minimisation problem [28]. The aim is to minimise the number of consecutive fixtures played at home or away in a schedule. Many leagues require this as it can affect a club’s cash-flow from gate revenue and fans may not be happy if they cannot watch their team at home for a long period of time. A league in which the number of breaks is optimal (i.e. $2n-2$ where $2n$ is the number of teams) is therefore considered desirable by league owners and attractive to fans. De Werra [28], through graph theory, completed a lot of pioneering work concerning breaks. The significance of this work led to the ‘first-schedule, then-break’ combinatorial scheduling methodology of finding a good *HAP Set* for a given *T Set* to form a schedule. Alternatively, the derivative of this approach is the reverse ‘first-schedule, then-break’ solving methodology (Section 3.3.4).

- **Carry-over effects minimisation**

Kendall et al. [28] describes the carry-over effect as follows, “It is said that team *i* gives a carry-over effect to team *j* if some other team plays consecutively against teams *i* and *j*”. For example, if team *i* is a very strong team (conversely a very weak team) then the opposing team could be physically fatigued and psychologically discouraged (conversely refreshed and encouraged) before playing team *j*, thereby benefitting (conversely hindering) team *j*. Therefore, the carry-over effects problem concerns fairness.

Goossens and Spieksma [35] developed an approach to measure whether carry-over effects have an influence on the outcome of fixtures. Over 30 seasons of data from the Belgium football league were used in the research. It was found that there was no evidence to support the claim that carry-over effects affect results and fairness. It was suggested that perhaps there is sufficient time between fixtures such that carry-over optimisation is unnecessary and does not improve fairness.

- **Group-changing and Group-balanced schedules**

A similar concept to the carry-over effect, with regards to fairness, is the strength team’s consideration. Here, teams are partitioned into different groups S based on strength ($|S|$ represents the number of groups). The objective is to then minimise the incidences in which a team has consecutive fixtures against extremely strong teams (or extremely weak teams).

Kendall et al. [28] highlights two concepts which have been introduced to group fairness. The first is called ‘group-changing’. Here, no team plays consecutive fixtures against an opposition from the same group S. The second is called ‘group-balanced’. This refers to the case where a team plays an opponent from every other group in consecutive slots.

Briskorn [36] provides proofs that a group-balanced 1RR can be arranged if and only if $n/|S|$ is even. Also, it is proved that there is no group-balanced 1RR where $|S|$ is odd. With regards to group-changing, it is proved that if $n/|S|$ is odd and $|S|>2$, a group-changing 1RR can be arranged.

- **Minimising Costs**

In the minimum cost problem, a cost c_{ijs} is incurred for scheduling team i to play against team j at the venue of team i in slot s for all teams and slots [28]. The objective is to find a feasible round-robin schedule satisfying several constraints with minimum total cost [28]. Briskorn [37] provides the basic minimal cost integer programming model for 1RR and 2RR and analyses its relationship to planar three-index assignments. It is also shown that the three-index assignment problem is useful in the 'first-break, then-schedule' decomposition models.

- **Balanced Tournament Design (BTD)**

In some sport scheduling problems, venues are a key consideration. A balanced tournament design of order n, BTD(n), is one in which $2n$ teams participate in a 1RR of $(2n-1)$ slots with n different venues. The effects of the different venues are said to be balanced when no team plays more than twice at any venue. This problem is called "prob026" in the CSPLib library [38]. Founded by Ian Gent and Toby Walsh in 1999, this library aims to provide standard problems for the constraint programming community [28].

This problem is particularly useful when balancing different kick-off times for teams over a season (can also be used for referee assignment). The problem can be represented by a matrix with columns corresponding to slots and rows to the kick-off timeslots (or referees) available [28].

Variants of the BTD include the factored balanced tournament design FBTD(n) and the partitioned balanced tournament design PBTD(n). A FBTD(n) is a BTD(n) in which each team plays in all venues in the first n slots. Furthermore, if each team then plays exactly once in each venue in the last n slots, it is then referred to as a PBTD(n) [28]. Dinitz and Dinitz [39] provides definitions and theorems regarding BTD's and also enumerate the designs for BTD(5), FBTD(5) and PBTD(5). This problem is applicable to scheduling tennis tournaments in order to balance the effects of courts on players.

- **Other sports scheduling problems**

Sports scheduling problems not discussed in the literature above include: time-relaxed problems (relaxed schedules), bipartite tournaments, referee assignment, and play-off or first place elimination. Literature references on the abovementioned problems can be found in the 'Classification of literature on sports scheduling' library [29] mentioned previously.

3.3.4 Scheduling techniques

Researchers and schedulers attempt to solve the various sports scheduling problems using numerous techniques. While some techniques work well for certain problems, they may not necessarily work well for others. As a result, researchers are challenged to identify the best techniques for the various problems. A number of exact and approximate scheduling techniques such as integer programming, constraint programming, decomposition approaches, heuristics, metaheuristics, and hybrid methods have been used to solve the various scheduling problems [28].

- **Decomposition**

Decomposition techniques are concerned with decomposing problems into sub-problems. These sub-problems are solved separately and then superimposed into a final solution. Two approaches have frequently been used in the literature.

'First-schedule, then-break'. Here, the problem is solved by first optimising a *T Set* according to some criteria like TV revenue maximisation. Afterwards, based on this fixed timetable, a feasible *HAP Set* which corresponds to the timetable is found if one exists. Briskorn [36] points out that a large focus of research is based on finding a feasible *HAP Set* with minimal breaks.

'First-break, then-schedule'. In this case, the *HAP Set* is first optimised. Often the objective is to minimise breaks in the *HAP Set* [28]. Afterwards, a corresponding feasible *T Set* is found if one exists.

Schreuder [40] constructs the real-world Dutch professional football league schedule using the first-break, then-schedule decomposition technique. The *HAP Set* is determined using edge-colouring (one-factorisation) of complete graphs.

Briskorn [36] formally presents the above decomposition methodologies using binary integer programming. Theorems and proofs to test feasibility are also provided. Alternatively, Kendall et al. [28] notes that constraint programming, complete enumeration, integer programming or heuristics algorithms can be used to solve these decomposition problems.

- **Integer Programming**

Sports scheduling problems can be formulated and solved using integer programming [28]. Integer programming is a common technique used in operations research. Also, it uses standard formulations which are easily programmable. As a result, there are numerous solvers available such as those provided in Matlab, CPLEX, SCIP and LINDO. Solvers use different solving schemes involving various cut and branching strategies. Therefore solvers have different performance characteristics.

Kendall et al. [28] formally defines the basic integer programming model used to schedule a 2RR as follows:

Variable definition where teams *i, j = 1,…,n* and slots *s = 1,...,|s|*.

$$x_{ijs} = \begin{Bmatrix} 1, & \textit{if team i plays at home against team j in slot s} \\ 0, & \textit{otherwise} \end{Bmatrix}, \quad i \neq j$$

The first constraint imposes that every team must play at home against every other team.

$$\sum_{s=1}^{|s|} x_{ijs} = 1, \qquad \forall 1 \leq i,j \leq n, \quad i \neq j$$

The second constraint imposes that in a compact schedule, no team plays more than once in each slot.

$$\sum_{j=1}^{n} x_{ijs} = 1, \qquad \forall 1 \leq \mathrm{i} \leq \mathrm{n}\,, 1 \leq s \leq |s|, \quad i \neq j$$

Kendall et al. [28] identifies branch and bound, branch and cut, Benders decomposition, and column generation as integer programming strategies used in sports scheduling literature. Problems solved using integer programming include real-world model development instances, the travelling tournament problem and the break minimisation problem [28]. Kendall et al. [28] notes that integer programming is commonly used in hybrid algorithms. In these cases, integer programming is often used to solve sub-problems. For example, it is used in decomposition scheduling techniques as well as in combination with constraint programming or heuristic scheduling techniques.

- **Constraint Programming**

Constraint programming combines the ideas of artificial intelligence with the developments of computer programming [41]. As a result, flexible programming allows for values to take on interrelated variables and constraints [41]. For instance, variables can take on a domain of possible numbers and constraints can be of different forms like mathematical, disjunctive, relational, explicit, unary, and logical constraints [41]. Constraint programming continuously fixes variable values and derives new constraints through the processes of domain reduction (eliminating possible values through logic conclusions) and constraint propagation (adjusting constraints) [41].

Schaerf [42] realises the power of constraint programming by formulating and solving sports scheduling problems. A two stage approach is used. The first stage concerns producing a schedule with dummy teams based on graph theory and the second stage assigns actual teams to the schedule. The first stage aims to minimise breaks and create complementary patterns and the second stage uses a constraint based depth-first branch and bound procedure to assign teams.

- **Heuristics and Metaheuristics**

Sports scheduling problems can be solved using standard search procedures such as integer programming. Such standard procedures may be computationally reasonable for smaller instances. However, as the size of the problem increases, the search region becomes too large and would require large processing power and time to solve using standard techniques. As a result, sport scheduling researchers look to find good solutions with the use of heuristic and metaheuristic techniques for larger and computationally hard instances.

Heuristics, designed specifically to solve a problem, are low-level search procedures used to solve large and complex problems. Heuristics search for, and investigate, local neighbourhoods by iteratively modifying the current solution and replacing it only if the local neighbourhood is a better one [28]. As a result, depending on the initial trial solution, this strategy will converge to non-global local optima unless the initial trial solution happens to be in the global optimum neighbourhood [41]. Generally, a heuristic finds suboptimal solutions in reasonable computational times.

Metaheuristics are similar to heuristics in that they too employ heuristic neighbourhood search strategies. The difference is that they orchestrate local neighbourhood moves based on higher level strategies [41]. This way, the orchestrated high-level search process guides the low-level heuristic to explore the feasible region (and in some cases, infeasible) by escaping from local optima. The metaheuristic then directs the heuristic to search the best avenues found in the region. As a result, this method finds good (often optimal) solutions.

The key advantage of heuristics is that they can easily be tailored to fit the problem at hand and reduce the mathematical and computational complexity. This is why researchers concerned with scheduling find such techniques to be the most effective in solving difficult combinatorial optimisation problems. Kyngäs [43] classifies and describes popular metaheuristics methods which have been used to solve various scheduling problems. These include ant colony optimisation, co-operative local search, ejection chains, genetic algorithms, hill-climbing, hyper-heuristics, iterated local search, memetic algorithms, particle swarm optimisation, simulated annealing, tabu search and variable neighbourhood search.

Hill-climbing is more of a basic local improvement procedure than a metaheuristic. It starts with a random initial solution and then continuously accepts small changes to the solution if the changes improve, or equal, a solution. The emphasis here is to "climb up the hill" (or downhill for minimisation problems) until a local optimum is reached. This procedure terminates when the solution can no longer be improved [43].

Simulated annealing always accepts improving moves but also allows for downhill moves. The probability of accepting downhill moves is decreased over time so that it finally diminishes to zero [43]. The acceptance of a move, for a maximisation problem, is governed by $exp(Z_n\text{-}Z_c/T)$, where Z_n is the trial solution cost, Z_c the current solution cost and T is a temperature control parameter which, over time, approaches zero. With higher temperatures, the emphasis is on diversification to unexplored regions by allowing for downhill moves to be accepted. At lower temperatures, the probability of accepting downhill moves is diminished with the emphasis shifting towards intensification. Finally at extremely low temperatures, the emphasis changes to hill-climbing [43].

Anagnostopoulos et al. [44] considers the travelling tournament problem of the salient features of Major League Baseball in the United States and proposes a simulated annealing algorithm. The algorithm explores both feasible and infeasible schedules and includes advanced techniques such as strategic oscillation and reheats to escape local optima at very low temperatures. For larger artificial instances, the algorithm produces significant improvements over previous approaches. Furthermore it is shown to be reliable since its worst solution over 50 runs was less than or equal to the best known solutions.

Kendall et al. [28] notes that in sports scheduling, often hybridisations of principles from different metaheuristics can produce the best results. It is also highlighted that metaheuristics strongly benefit from good initial solutions which can be obtained by greedy randomised constructive heuristics.

Lim et al. [45] uses a hybrid metaheuristic approach in scheduling the travelling tournament problem. Simulated annealing and hill-climbing are used in conjunction to create a schedule. Here, a timetable is found using simulated annealing. The best timetable found is then passed to the hill-climbing component which searches for better team assignments. The best schedules are then reverted to the simulated annealing component. This process terminates when there is no improvement for a specified number of iterations or when a time limit is reached. Using benchmarks, the results presented are comparable to the best known solutions.

- **Other sports scheduling techniques**

Other sports scheduling techniques in the literature not described include SAT formulation, semi-definite programming and Lagrange relaxation. Literature references on the abovementioned problems can be found in the 'Classification of literature on sports scheduling' library [29].

3.3.5 Local neighbourhood search for sports schedules

As stated previously, heuristics are used to find solutions in a search region when traditional techniques become computationally unreasonable. Local search procedures find feasible moves into the local neighbourhood from the current solution [41]. Anagnostopoulos et al. [44] and Ribeiro and Urrutia [46] present neighbourhood search strategies which are commonly found in the sports scheduling literature.

Anagnostopoulos et al. [44] proposes several neighbourhood search strategies to solve an artificial travelling tournament problem. The five strategies outlined in the article are listed:

a. SwapHomes: Switches the venue roles of the two fixtures between two arbitrary teams in a schedule (for a 2RR).

b. SwapRounds: Switches an arbitrary slot in a schedule with another arbitrary slot in the schedule.

c. SwapTeams: Switches the set of fixtures of an arbitrary team with the set of fixtures of another arbitrary team. The schedule is then updated to form a feasible schedule.

d. PartialSwapRounds: This move swaps the fixture of an arbitrary team in an arbitrary slot with the fixture of another slot. The rest of the schedule is then updated using ejection chains.

e. PartialSwapTeams: In an arbitrary slot, the fixtures of two teams are switched. The rest of the schedule is updated using ejection chains.

Anagnostopoulos et al. [44] state that the first three neighbourhood search strategies are incapable of sufficiently exploring the feasible region.

Ribeiro and Urrutia [46] defines four neighbourhood strategies which were used to solve the mirrored travelling tournament problem. Identical strategies to SwapHomes, SwapTeams and PartialSwapRounds are presented in addition to a strategy called GameRotation. Here, an arbitrary fixture is forced to be played in an arbitrary slot. The schedule is then updated using complex ejection chains. Importantly, the authors notice that there are moves in the PartialSwapRounds and GameRotation which the solutions generated by the polygon method for $n = 8,12,14,16,20,24$ cannot produce. This is because this strategy is able to change the structure of the one-factorisation built by a polygon construction method. The polygon method is described in Section 3.3.7.

Ribeiro and Urrutia [46] states that that there are $n(n-1)/2$ possible moves for each of the SwapHomes and SwapTeams in a 2RR. Also, it is pointed out that the partial swap moves may only work if there are two oriented one-factors for which there are four common vertices which are connected. Furthermore, after the partial swap, there are 16 combinations of possible venue assignments which must be evaluated.

3.3.6 Football Application

A wide range of sports scheduling applications exist. Kendal et al. [28] identify articles applicable to football, basketball, cricket, baseball, ice hockey, tennis, table tennis, golf pairing and referee assignment. A number of exact and approximate optimisation methods have been widely applied to football scheduling problems [28]. Some are shown here.

Bartsch et al. [31] schedule the real-world German and Austrian Football Leagues. Here, a semi-greedy heuristic algorithm and a branch-and-bound algorithm are compared with the semi-greedy algorithm preferred. The authors worked with the league authorities in each league and by the time the article was published, the schedules generated using the method developed were used by the league owners on one occasion for the German league and on six occasions for the Austrian league.

Della Croce and Oliveri [47] attempt to schedule the Italian Football League. The proposed procedure is divided into three phases, all of which are solved using integer programming. The first phase generates a *HAP Set* respecting the broadcaster's requirements and several other constraints. A feasible *T Set* compatible with the *HAP Set* is found in the second phase. Lastly, the schedule is generated by assigning teams to the timetable. It is stated that the authors were in contact with the Italian Football League for the possible application of the proposed algorithm.

Kendall [4] attempted to schedule the fixtures over the holiday period of the EPL (a dissimilar case to the TTP). Here, the objective was to minimise the total distance travelled by all teams over the Boxing Day and New Year's Day fixtures (due to bad weather, fan consideration and limited public transport and policing resources). According to Kendall [4], the league had trouble with this aspect of their schedule. Depth-first integer programming, followed by a local search algorithm was used. In the results, it is shown that a 25% improvement on the official schedules was obtained over three seasons.

Kendall [4] notes that the league had usually scheduled these fixtures first and then scheduled the rest of the season around it. A limitation to this, however, is that although this approach may generate optimal distances travelled over these two slots, it may negatively impact other constraints within the season. Nevertheless, Kendall [4] has discussed the findings with the Football League authorities and is currently working closely with the league. The work is subject to a non-disclosure agreement.

Goossens and Spieksma [48] describe how they improved the Belgian football league using operations research after the league authorities had invited them to work on scheduling the upcoming leagues. Up until the 2005/06 season, a calendar committee was assigned to construct the schedule by hand. After being invited to automate the system, the authors formulated a first-break, then-schedule problem and solved it using integer programming. The schedule developed was used for the 2007/08 season.

3.3.7 Basic graph theory and the polygon construction method

- **Basic theory**

A graph G in which every vertex is joined by an edge to every other vertex is called a complete graph and is denoted by K_n. A complete graph can be viewed as a 1RR tournament. Wallis [49] provides an example of the complete graph K_6, representing a 1RR for six teams in Figure 1. A one-factorisation of a complete graph can then be used to schedule a 1RR tournament. Wallis [49] illustrates the one-factorisation of K_6, shown in Figure 2, where each one-factor graph represents a slot of fixtures. Here, each edge between the adjacent vertices represents a fixture. Note that there are five one-factor graphs in the one-factorisation of K_6, which corresponds to a 1RR of six teams where there are five slots (i.e. *n-1*).

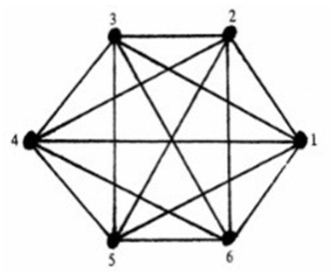

Figure 1: Complete graph K_6 [49]

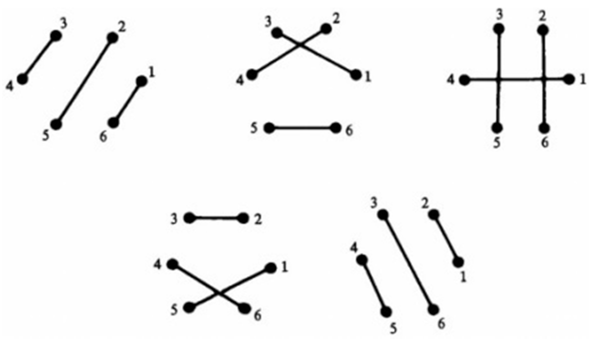

Figure 2: One-factorisation of the complete graph K_6 [49]

- **Polygon construction**

Commonly, the well-known polygon construction method is used to systematically schedule a 1RR tournament based on one-factors. Ribeiro and Urrutia [46] describe the polygon method:

A regular polygon graph of *(n-1)* vertices is initially drawn with a centre vertex also drawn. Teams labelled *1,...,(n-1)* are consecutively placed clockwise on the vertices of the regular polygon graph with the centre labelled as team *n* (see Figure 3).

Horizontal lines (edges) are then drawn between the vertices with the remaining vertex drawn to the centre. These edges represent a one-factor of the complete graph K_n and correspond to the fixture list of the first slot. The second slot of fixtures is found by rotating the polygon clockwise to the next vertex with new edges automatically formed. For n teams, there will be $(n-1)$ rotations (i.e. slots). This method aids in attaining the one-factors in the one-factorisation of a complete graph in an ordered way, especially for large instances.

Figure 3 illustrates how the schedule is created for an even number of teams ($n = 8$) using the polygon construction method. For an uneven number of teams, team $(n+1)$ represents a bye.

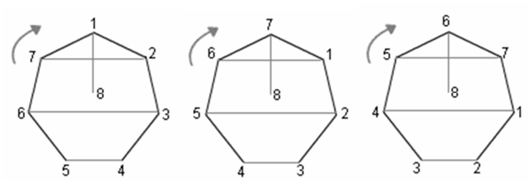

Figure 3: Polygon method for $n = 6$ (three rounds)

To obtain the venues of fixtures in the one-factors, the regular polygon edges can be converted into a digraph. Here, the arrow-head of the horizontal edges from left to right and the vertical edge from top to bottom can represent the away team.

3.4 Benchmarks

Nurmi et al. [3] attempts to standardise a framework for a highly constrained sports scheduling problem. The framework was created by observing the objectives and constraints of various professional sports leagues from around the world. A set of artificial instances are proposed and a set of real-world instances are presented (Austrian, Chilean, Belgium, German and Brazilian football leagues). Best known solutions for each set are published. This way, researchers can test the value of their scheduling techniques.

- Artificial instances

Artificial instances in the literature are concerned with either the travelling tournament problem or the break minimisation problem (see Section 3.3.3). The travelling tournament problem is frequently used for leagues with a large geographical spread of teams, like Major League Baseball in the United States.

- Real-world football instances

Football leagues around the world usually have their own set of distinct constraints which aim to increase the quality of the league and meet stakeholder needs. Some constraints of real-world football instances presented by Nurmi et al. [3] include: scheduling specific games to a subset of slots (Austrian), a subset of teams cannot be scheduled to play an arbitrary number of home games in the same slot (German), teams which share home venues cannot be scheduled home fixtures in the same slot (Belgian and Italian), avoid carry-over effects (Belgian), spread attractive games across the season (Belgian) and schedule attractive games to weekend slots to increase stadium attendance and TV viewership (Brazil).

In each of the abovementioned leagues, the most important goal is to minimise the number of breaks. In comparison, this goal is relaxed in the EPL scheduling problem with a maximum of two consecutive home or away fixtures permitted. Also, the Austrian, German, Belgian and Brazilian leagues are mirrored 2RR instances whereas the EPL is simply a 2RR instance. Unlike the football leagues mentioned, the EPL season continues throughout the Christmas period. It is therefore a requirement that the travelling distance over the holiday period is minimised. Therefore, the EPL problem is a distinct case to other real-world football sports scheduling problems available in the literature which includes those artificial and real-world benchmark instances as presented by Nurmi et al. [3].

- Unsolved instances

Nurmi et al. [3] has invited the scheduling community to formulate and solve instances not already available in the literature such as the real-world EPL instance in this research. To be consistent with the standards set by Nurmi et al. [3], this research will adopt the constraint framework proposed in that article (used in Section 5.3).

3.5 Complexity

P vs. NP is an unsolved mathematical problem which pertains to the relationship between the solving time and complexity of problems. P is the class of problems which is solvable using known algorithms in polynomial time. NP is the class of problems whose solution can be verified in polynomial time but is not solvable in polynomial time using known algorithms. The unsolved problem concerns whether P = NP, in which case problems whose solution can be verified in polynomial time can also be solved in polynomial time, or P ≠ NP, in which case there are problems which are harder to solve than its solution verified. The official problem description can be found on the Clay Mathematics Institute's Millennium Problems page [50].

Briskorn [36] provides proofs that the 1RR, mirrored 2RR and 2RR are all NP-complete and in so being, implies that they are in the NP-hard computational complexity class. Therefore, as the instances consisting of an increasing number of participating teams increase, the solution cannot be solved in polynomial time using exact scheduling techniques currently known. Instead, it is exponential, and in some cases, faster than exponential.

The logarithm *ln(t)* line graph in Figure 4 (*t* is time in seconds and *n* is the number of teams), provided by Briskorn [36], illustrates the run time behaviour of basic 1RR instances solved using integer programming via CPLEX. This corresponds to the NP-hard proofs provided.

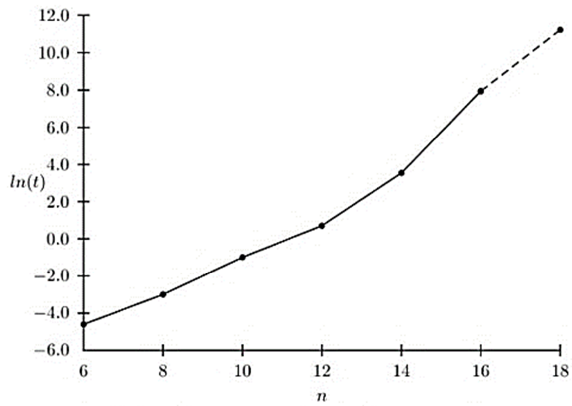

Figure 4: Run time behaviour for 1RR instances [36]

3.6 Literature survey summary

Relevant data in terms of the different tournaments accompanying the EPL, stakeholder requirements and revenue sources have been identified. All of which can influence the schedule. For example, a team should have a break around a FAC fixture. Also, stakeholder requirements may conflict with one another but hard constraints must be satisfied. For example, fans want to travel a minimal distance on Boxing Day and New Year's Day but clubs do not want to play local rivals on these days. A scheduler must therefore minimise travel distance on these days but cannot consider local rivals which would otherwise optimise travel distances. In the absence of the relative importance of different requirements, revenue of each tournament is useful. The potential revenue gained per tournament can be used to relatively prioritise the requirements of tournaments and stakeholders.

A Fixtures Working Party in conjunction with Atos constructs the EPL schedule each year using a non-disclosed software methodology and scheduling technique developed by Atos. However, identifying with the current macro methodology followed, it appears that it is one that can be classified as a first-break, then-schedule scheduling technique.

At the time of writing, Kendall [4] provided the only article on scheduling the EPL. However, the article only considered optimising the distance travelled over the Boxing Day and New Year's Day slots which he claims the league had trouble with. In an email contact [51], to further discuss his communication with the Football League, he said that he was now working with the league and the work is subject to a non-disclosure agreement.

Benchmarks do exist for both artificial and real-world instances and are maintained by researchers. However, these instances are mostly concerned with the travelling tournament problem or the break minimisation problem. The EPL does not require either. Nurmi et al. [3] encourages the scheduling world to solve unsolved problems like the EPL. This is what this research will attempt.

Finally, to place the complexity of the EPL sports scheduling problem in context, it was found in the literature that 2RR instances are NP-complete and therefore NP-hard. As a result, metaheuristics are likely to be the best scheduling technique known to solve the EPL scheduling problem.

4 METHODOLOGY

Once the relevant EPL information, sports scheduling problems and solving techniques were identified from the literature, a methodology was developed (flowchart form in Figure 5). At a high-level, the first process concerns data acquisition. This is followed by scheduling the EPL using a metaheuristic solver. The solver was then verified and the results were validated.

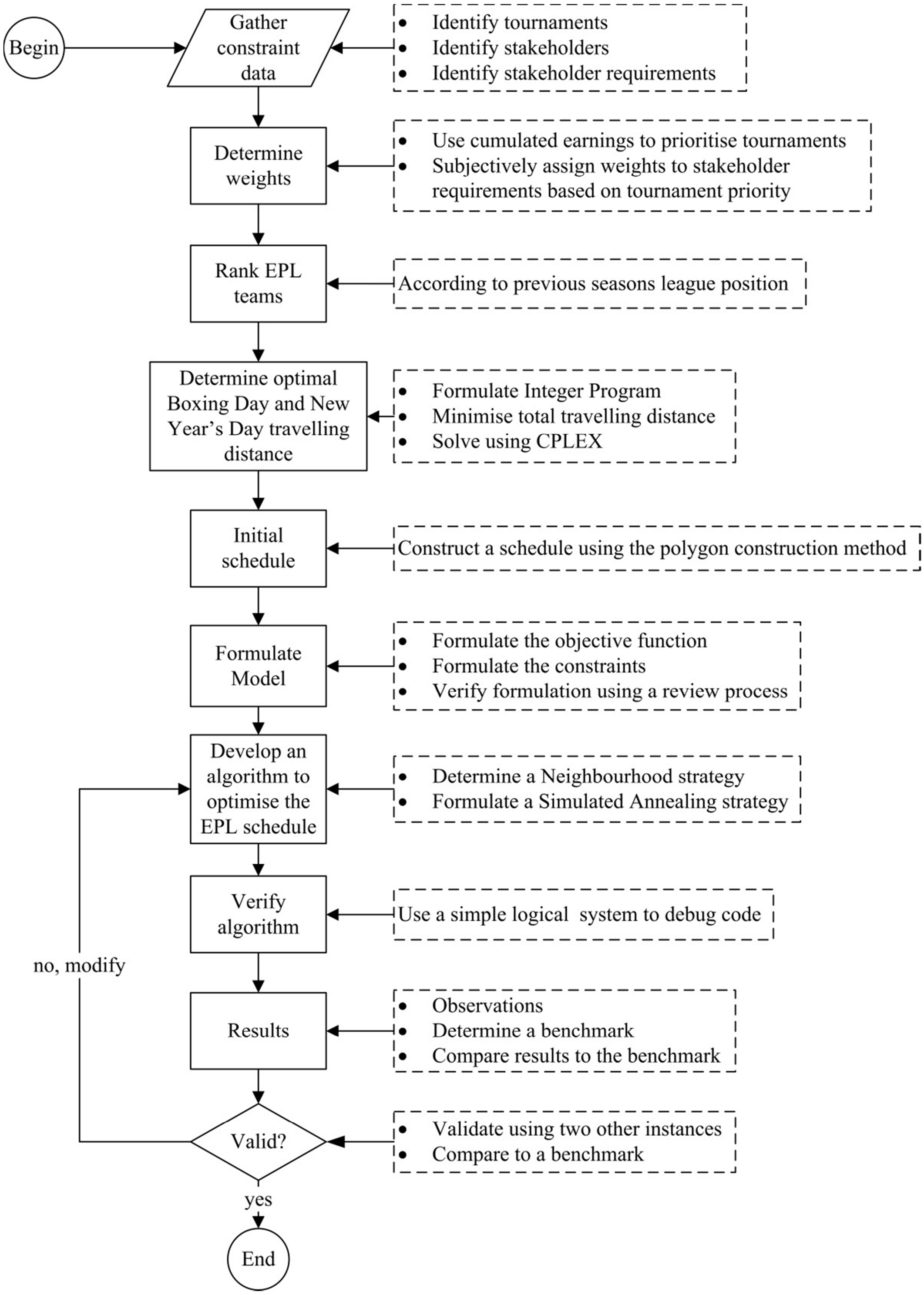

Figure 5: Methodology

5 EPL SCHEDULING PROBLEM

5.1 Introduction

There are many football scheduling applications within the sports scheduling problem literature which range from artificial to real-world instances. Although benchmarks exist for the Austrian, Chilean, Belgium, German and Brazilian leagues in Nurmi et al. [3], a benchmark for the EPL does not. The EPL scheduling problem is a substantially different case compared to other football leagues (see Section 3.4).

Scheduling the EPL without considering the various stakeholders is inadequate. The clubs, league, police and fans and all their requirements and constraints need to be considered. All requirements should be consolidated in the EPL scheduling problem in order to provide an effective real-world solution. To be consistent with sports scheduling literature, the constraints used in Section 5.3.1 are defined according to Nurmi et al. [3]. Additional constraints unique to the EPL which are not considered in the framework are included.

- **Instance**

Specifically, the EPL instance is a highly-constrained 20 team $(n = 20)$ compact 2RR. Therefore a total of $n(n\text{-}1)$ fixtures are to be scheduled into $2(n\text{-}1)$ slots of fixtures.

- **Problem Scope**

The scope for EPL scheduling will include the two European cups (UCL and UEL) and the domestic FAC for the 2013/14 season. The justification is that these are the potentially lucrative competitions which teams wish to compete for (see Section 3.1.2). Therefore, the FLC requirements will be omitted since the potential earnings from this competition are small in comparison and there are no specific FLC requirements to be considered. In the rare case a team qualifies for the UEFA Super Cup and Club World Cup, this team would have to compete in three fixtures which interfere with the EPL schedule. In this case, the EPL league organisers would minimise interference with the league by postponing league fixtures (usually to midweek). Therefore, these competitions are also not considered in the problem.

5.2 Objective Function (Z)

The objective of the EPL scheduling problem is to identify an optimal feasible schedule that violates no hard constraints while minimising the total cost incurred by soft constraint violations. A total Z-cost in excess of a Big-M weight (described in Section 5.3.2) suggests that a hard constraint is violated. In this event, the solution is either highly undesirable or even infeasible and should not be considered. Conversely, a Z-cost close to zero is highly desirable since it violates no hard constraints and has minimal soft penalty infringements.

$$Z_{min} = \sum \text{weighted constraint penalties}$$

(Such that cumulated hard constraint penalties = 0)

5.3 Constraints

Constraints which have been considered for the EPL scheduling problem are listed next (Section 5.3.1) and the weights, which distinguish between relative soft and hard constraints, follow (Section 5.3.2).

5.3.1 Constraint definitions

Constraints considered include those which address the various stakeholder requirements as well as those which are required to force feasibility of solutions. Stakeholder requirements include those from the league, clubs, police and fans. Feasibility constraints ensure that the solution generated is feasible with regards to a 2RR. Additional requirements not found in the literature have been included in this research to improve the probability of EPL teams succeeding in lucrative European tournaments whilst not compromising their league aspirations.

- **Home and Away venue constraints**

C04[2]. Team *t* should not play at home in slot *s* (e.g. policing and club request due to some other event happening in close proximity to the stadium).

C07. There should be at least m_1 and at most m_2 home games for teams t_1, t_2 … in the same slot (e.g. $m_1 = 0$ and $m_2 = 1$ to ensure paired teams do not play at home in the same slot).

[2] Constraints with a C code in the region over C100 are constraints created by the author specifically for the EPL scheduling problem while the rest are those found in Nurmi et al. [3].

- **Strength and Paired team constraints**

C102. Teams in the strength group should not be assigned to compete in slot *s* (C102 class listed alphabetically below).

C102a. Avoid strength fixtures on the first game of the season (i.e. slot s_1).

C102b. Avoid strength fixtures within the holiday period (i.e. Boxing Day slot s_{18} and New Year's Day slot s_{20} for the 2013/14 season).

C102c. UCL teams should not be assigned a strength team in the EPL prior to a UCL fixture.

C102d. UEL teams should not be assigned a strength team in the EPL prior to a UEL fixture.

C102e. UCL teams should not be assigned a strength team in the EPL which succeeds a UCL fixture.

C102f. UEL teams should not be assigned a strength team in the EPL which succeeds a UEL fixture.

C103. Paired teams should not be assigned to slot *s* (i.e. holiday period fixtures in slots s_{18} and s_{20}).

- **Break constraints**

C12. A break cannot occur in slot *s* (C12 class listed alphabetically below).

C12a. Avoid a break on the first two slots of the season (i.e. slots s_1 and s_2).

C12b. Avoid a break on the last two slots of the season (i.e. slots s_{37} and s_{38}).

C12c. Teams cannot have consecutive home (or away) holiday period fixtures (slots s_{18} and s_{20}).

C13. Teams cannot have more than k consecutive home games (i.e. $k = 2$).

C14. Teams cannot have more than k consecutive away games (i.e. $k = 2$).

C105. EPL matches either side of a FAC fixture should not both be away from home.

C106. Each team should have an equal number of home and away Bank Holiday fixtures, assuming an even number are scheduled.

- **Tournament quality constraints**

C104. Teams must play every other team before a fixture can be repeated in the 2RR (i.e. a fixture in the first half of the season can only be repeated from the second half onwards – from slot s_{20} onwards).

C107. The total travelling distance over the holiday period fixtures should be minimised to lessen the inconvenience to travelling fans (i.e. the total travelling distances between stadiums for all fixtures in slots s_{18} and s_{20} should be minimal).

C108. The holiday period fixtures on Boxing Day should not be repeated on New Year's Day (i.e. a fixture in slot s_{18} cannot be repeated in slot s_{20}).

C19. There should be at least k slots between two fixtures with the same opponents (i.e. in order to obtain a balanced spread of games, fixtures cannot be repeated in the period between s_{14} and s_{25} which accounts for six games either side of the 2RR halfway mark).

- **Feasibility constraints**

C110. The sum of each entry in row t (teams) in the binary *HAP Set* of a compact 2RR must equal $|s|/2$ (i.e. for $n = 20$ in a 2RR, there are $|s| = 38$ slots, therefore $|s|/2 = 19$ since there will be 19 fixtures at home (1) and 19 away (0)).

C111. The sum of each entry in column s (slot) in the binary *HAP Set* of a compact schedule must equal $n/2$ (i.e. for $n = 20$ teams, $n/2 = 10$ since there will be 10 fixtures per slot with 10 teams playing at home (1) and 10 away (0)).

C112. The sum of each absolute entry in row t (team) in the timetable *T Set* of a compact 2RR consisting of n teams must equal $2(\sum n)-2t$ (i.e. for $n = 20$ and $t = 1$, $2(\sum n)-2t = 418$).

C113. The sum of each absolute entry in column s (slot) in the timetable *T Set* consisting of n teams in a compact schedule must equal $\sum n$ (i.e. for n = 20, the sum total of $\sum n = 210$).

5.3.2 Weight assignment

Hard and soft constraints describe the relative importance of constraints. Hard constraints are those which *must* be satisfied to ensure a feasible solution while soft constraints are *criteria* which make the feasible solution more attractive to stakeholders.

Weights are assigned to decipher the relative importance of constraints – the larger the weight, the more important it is. Hard constraints receive large weights or Big-M weights so that the likelihood of a solution containing a hard violation is highly unlikely. Conversely, soft constraints are assigned smaller weights which allows for some violations. Soft constraints were set subjectively[3] according to potential earnings of each tournament. These weights were then altered by trial and error to find a set of weights which appeared fair and produced the best results. Table 5 shows how the weights were categorised.

Table 5: Weight range

Tourn.	Approx. value**	Cumulative value	Range used
FAC	€ 4m	€ 4m	00-02
UEL	€ 11m	€ 15m	02-09
UCL	€ 60m	€ 75m	09-45
EPL*	€ 100m	€ 175m	45-100

* league winners broadcasting revenue for the 2013-2016 cycle
** based on EUR-GBP of 1.14 (31 Aug 2013)

The approximate earnings, or value, of each tournament are those found in Section 3.1.2. The range used was defined by calculating the approximate contribution of each tournament with regards to the total earnings. For example, the UEL value of the total potential earnings is *€11m/€175m ≈ 0.07 (7%)* - hence the weight range is defined from 02 to 09 (i.e. *02+07 = 09*). This means that the UEL is more important than the FAC but not as important as the UCL and EPL [4]in terms of value. Table 6 illustrates the subjective weights assigned to each constraint. In the table, constraints are listed in order of appearance in Section 5.3.1.

[3] In the absence of real weights assigned to constraints by the Fixtures Working Party, a subjective approach is used instead. In reality, these weights are decided based on negotiations between stakeholders.
[4] Only the final league standing portion of the broadcasting revenue is considered since the remaining portion is distributed evenly amongst all the EPL teams regardless of performance.

Table 6: Weight assignment

Constraint	Tourn. applicable	Assigned weight	Constraint	Tourn. applicable	Assigned weight
C04	EPL	60	C13	EPL	100
C07	EPL	100	C14	EPL	100
C102a	EPL	80	C105	FAC	2
C102b	EPL	90	C106	EPL	50
C102c	UCL	45	C104	EPL	1 000
C102d	UEL	5	C107	EPL	0.5/km
C102e	UCL	30	C108	EPL	1 000
C102f	UEL	9	C19	EPL	60
C103	EPL	100	C110	EPL	M*
C12a	EPL	90	C111	EPL	M
C12b	EPL	90	C112	EPL	M
C12c	EPL	1 000	C113	EPL	M

* M represents Big-M, a large penalty, with a weight of 1×10^6

Weights are assigned in Table 6 within the ranges presented in the weight range table (Table 5). Less important constraints receive weights closer to the lower end of the range while more important constraints receive values on the upper end of the range. For example, consider C102c which aims to minimise the number of strength fixtures prior to a UCL fixture. It falls within the 09-45 range in the weight range table (Table 5). Furthermore, C102c is an important requirement for teams participating in the UCL and therefore this particular weight receives a weight in the upper range (i.e. 45).

There are three exceptions to the weighting rule described above:

- Constraints C12c, C104 and C108 have assigned weights, through trial and error, of 1000. If these three constraints are violated in the final solution, the schedule would be highly undesirable. However, a Big-M weighting is not used in order to allow the algorithm to temporarily violate these constraints in the search for highly attractive solutions.
- Constraints C110, C111, C112 and C113 have been assigned Big-M weights since these constraints must not be violated at any time. If they are violated, the schedule would not produce a feasible 2RR.
- Constraint C107 is unique since it is a measure of the gap between a solutions distance against the known optimal for the holiday period fixtures. Through trial and error, it was found that a penalty weight of 0.5/km was ideal. That is, this weight was not too large to influence the solution in producing a schedule that highly favoured this constraint. Also, it was not too small to produce a schedule which disregarded this constraint.

5.4 Parameters

The parameters used in the EPL scheduling problem are presented and described below.

- **Team parameters**

Table 7 shows the team parameters for the 2013/14 season.

Table 7: Team parameters

Team	Team code	Paired team/s	Strength teams	UCL	UEL
Manchester Utd.	t_1	t_2	•	•	
Manchester City	t_2	t_1	•	•	
Chelsea	t_3	t_{12}	•	•	
Arsenal	t_4	t_5	•	•	
Tottenham	t_5	t_4	•		•
Everton	t_6	t_7	•		
Liverpool	t_7	t_6			
West Brom. Albion	t_8	t_{15}			
Swansea City	t_9	t_{19}			•
West Ham Utd.	t_{10}				
Norwich City	t_{11}				
Fulham	t_{12}	t_3			
Stoke City	t_{13}				
Southampton	t_{14}				
Aston Villa	t_{15}	t_8			
Newcastle United	t_{16}	t_{17}			
Sunderland	t_{17}	t_{16}			
Hull City	t_{18}				
Cardiff City	t_{19}	t_9			
Crystal Palace	t_{20}				

Team code is an alphanumeric value assigned to teams based on the previous seasons final league placing. For the three promoted teams, these teams receive the last three team codes.

Paired teams are those teams which are in close geographical proximity. Teams which are paired in the table were those identified in Section 3.2. Although not in this case, a team can be paired to more than one team.

Strength group S represents the best teams in the EPL league. Teams could be ranked based on a performance centred co-efficient index over a period of time. Instead, since the strength of teams can change every season, it will be assumed that teams finishing the previous season in the top six positions (i.e. those challenging for the lucrative UCL and UEL positions) form the strength group for the following season. With the inclusion of Liverpool, the strength teams shown here are generally regarded as the best teams.

The *UCL group* consists of those teams which have qualified for the UCL. Since the EPL schedule is published before the UCL preliminary rounds begin, it is assumed that the teams competing in these preliminary rounds will qualify for the competition. Due to the strength of the EPL teams, it is rare that they do not qualify for the UCL.

The same applies for the *UEL group* as the UCL group. It is important to note that Wigan, who won the 2012/13 FAC and thereby qualified for the 2013/14 UEL, was relegated from the EPL for the 2013/14 season. Therefore, since Wigan is not in the 2013/14 EPL league, only Tottenham and Swansea, who were the other UEL qualifying teams for the 2013/14 season, and are in the 2013/14 EPL league, are considered in this problem.

- **Slot parameters**

Table 8 superimposes the dates of competitions and certain constraints with the respective slots of the EPL schedule. Only the slots which have an event are shown. Actual dates can be seen in Appendix B.

Table 8: Slot parameters

	s_2	s_3	s_4	s_6	s_8	s_{10}	s_{12}	s_{15}	s_{18}	s_{20}	s_{22}
FAC										•	•
UEL			•	•	•	•	•	•			
UCL			•	•	•	•	•	•			
C04	$t_1,t_2,$ t_3,t_{12}	t_{17},t_{19}	t_{16},t_{17}	t_1,t_2							
C106	•								•	•	

	s_{26}	s_{27}	s_{29}	s_{30}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}
FAC	•		•				•			
UEL	•	•	•	•	•	•		•	•	
UCL	•	•	•	•	•	•		•	•	
C04						$t_3,t_6,$ t_7,t_{12}				
C106								•		•

The dates for *FAC* fixtures are necessary for constraint C105. Here, since these fixtures are usually between two EPL slots, a black dot represents the slot leading to a FAC fixture. Likewise, the dates for the *UEL* and *UCL* fixtures are needed for constraints C102c to C102f and are also between slots. Constraint C04 concerns requests for away fixtures of some teams for certain dates. These have been superimposed with the respective slots. C106 superimposes the Bank Holiday dates with the respective slots.

- **Distances**

Distances between each team (measured in km) were found by determining the relative driving distance between each stadium (Table 9). This was done with the aid of Google Maps. Stadiums used can be seen in Appendix C. The diagonal has been assigned a Big-M distance of 1×10^4 km since a team cannot compete against itself.

Table 9: Driving distances between stadiums [km]

Team	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
t_1	M	8	339	321	325	57	53	135	379	341	392	338	81	368	141	243	240	167	319	352
t_2	8	M	339	321	323	62	58	133	377	339	314	338	80	367	141	229	226	153	319	352
t_3	339	339	M	13	22	355	354	210	300	20	195	3	260	125	202	451	448	347	241	14
t_4	321	321	13	M	8	337	336	195	312	15	180	18	250	136	184	450	447	337	251	21
t_5	325	323	22	8	M	344	340	200	318	19	174	23	253	152	187	451	447	303	258	26
t_6	57	62	355	337	344	M	1	154	398	360	411	358	100	385	157	282	279	203	335	368
t_7	53	58	354	336	340	1	M	150	394	356	407	353	97	383	156	278	275	203	335	367
t_8	135	133	210	195	200	154	150	M	246	214	266	83	64	238	6	341	338	230	186	223
t_9	379	377	300	312	318	398	394	246	M	319	500	300	307	284	251	579	575	469	69	313
t_{10}	341	339	20	15	19	360	356	214	319	M	174	23	269	145	204	450	447	303	259	25
t_{11}	392	314	195	180	174	411	407	266	500	174	M	204	322	327	255	414	412	245	441	200
t_{12}	338	338	3	18	23	358	353	83	300	23	204	M	267	124	201	458	455	346	240	16
t_{13}	81	80	260	250	253	100	97	64	307	269	322	267	M	297	70	315	238	204	248	281
t_{14}	368	367	125	136	152	385	383	238	284	145	327	124	297	M	229	521	517	408	224	147
t_{15}	141	141	202	184	187	157	156	6	251	204	255	201	70	229	M	330	327	217	191	215
t_{16}	243	229	451	450	451	282	278	341	579	450	414	458	315	521	330	M	24	229	520	473
t_{17}	240	226	448	447	447	279	275	338	575	447	412	455	238	517	327	24	M	225	517	470
t_{18}	167	153	347	337	303	203	203	230	469	303	245	346	204	408	217	229	225	M	407	360
t_{19}	319	319	241	251	258	335	335	186	69	259	441	240	248	224	191	520	517	407	M	254
t_{20}	352	352	14	21	26	368	367	223	313	25	200	16	281	147	215	473	470	360	254	M

6 MODEL DEVELOPMENT

This section describes the solving methodology and technique used to generate the EPL schedule. The model was created using the research literature highlighted in Section 3.3.

An overview of the model is first described to gain a high-level perspective. This is followed by the low-level algorithm formulated to construct the schedule. Lastly, the verification process followed in this research is described.

6.1 Model Overview

6.1.1 High-level overview

At a high-level, the model can be viewed in three parts:

1. The first part of the model identifies the optimal holiday period (Boxing Day and New Year's Day) travelling distance using a binary integer program. This is used as a reference to which a candidate solution can be compared to in the simulated annealing algorithm.

2. The second part of the model uses a simulated annealing metaheuristic algorithm (called the 'SA solver') to iteratively identify and compare schedules. Here, a candidate schedule is identified using a neighbourhood search heuristic. It is then deconstructed into a *HAP Set* and a *T Set*[5] from which the total cost can be calculated.

3. The third part of the model is the incorporation of a hill-climbing algorithm. Once the simulated annealing algorithm outputs a schedule, the hill-climber ensures that the local optimum in a promising search area is found.

[5] To reiterate, the *HAP Set* is the sequence of home and away fixtures for each team and the *T Set* is the sequence of opposition allocation for each team.

6.1.2 Model scope and assumptions rationale

Model scope and assumptions may affect the solution quality and validity of the results. The scope and assumptions are described here while the validity is tested through a validation process which is presented later in Section 8.3.

Scope concerns the competitions considered, the stakeholders considered and the stakeholder needs considered, all of which influence the EPL solution quality. Similarly, assumptions can influence the quality of a schedule and may originate from the scope or derive from the simplification, formulation and development of the model. The rationale for the model scope and assumptions made are described in detail:

- **Competitions considered**

Competitions considered include the UCL, the UEL and the FAC. This entails scheduling ‘kind’ fixtures (i.e. fixtures against non-strength teams) around European fixtures for participating teams and scheduling alternating sequences of home and away fixtures around predefined FAC rounds. Competitions not considered include the FLC, the UEFA Super Cup, the Club World Cup and the UEFA and FIFA international fixtures.

With regards to European cup fixtures, it is assumed that the UEL and UCL teams participating in the preliminary rounds will qualify for the competition. Since the league begins before the preliminary rounds are complete, this assumption must be made.

The needs of the domestic FLC is not considered on the basis that it is seen as a ‘feeder’ competition for youth and fringe players as a result of the comparatively small prize earning potential of the competition. These fixtures are scheduled by the Football League and do not interfere with the EPL.

In the event of a team competing in the Club World Cup, EPL fixtures for that team will be postponed by the league authorities to available dates (no team qualified for this competition in the 2013/14 season). UEFA and FIFA international fixture days are also not considered on the basis that these dates are ruled out before the EPL schedule is distributed to available dates by the Premier League administration.

- **Penalties and weights**

Weights of penalties assigned to the respective competitions were defined in Section 5.3.2. The weightings are in proportion to the relative prize earnings of each competition (i.e. weight: EPL>>UCL>>UEL>>FAC).

A conservative penalty approach is favoured in this model over an approach which may underestimate progress. Therefore, penalties in this model will be incurred even if teams do not reach the latter stages of competitions (European and FAC). This is because it is impossible to predict how far each team will progress in a competition. For example, say two UCL strength teams were assigned to play strength fixtures towards the end of a season prior to a UCL fixture, penalties would have been incurred because constraint C102c was violated. However if both these teams had been knocked out of the UCL by then, these penalties would be needless.

Considering that the importance of the EPL>>UEL, it is more important for the schedule to consider the EPL needs than a UEL team's needs. As a result, penalties for constraints C102d and C102f are only assigned if the participating UEL team happens to be a strength group team.

- **Football divisions**

Some EPL teams are paired with teams in the lower divisions (under the Football League administration). This can particularly influence the *HAP Set*. Since the EPL is the most lucrative league in the world, it is assumed that the lower divisions can be scheduled around the EPL. For this reason, the Football League will not be considered in the formulation of the EPL and is beyond the scope of this research.

- **Paired teams and strength teams**

It is assumed that the paired teams and strength teams shown in Section 5.4 are accurate and will be accepted by the league authorities.

- **Away requests and transport routes**

Highlighted in the literature (Section 3.2), clubs may request an away fixture on certain dates due to events happening in close proximity to their stadium. Policing constraints may include minimising home fixtures for paired teams in the same slots, which may otherwise stretch local services (policing, medical and public transport).

Due to a lack of disclosed information with regards to club or policing requests, the model scope was set around available information. The solution quality is reduced if those considered here are not considered by the league. Conversely, the quality is reduced if there are important constraints which are not considered here. However, a programmer could easily access the Matlab code used to create the schedule and translate additional constraints into code.

Transport ‘pinch points’ were not considered in the model. Pinch points concern the congestion of public routes when a large fan base from multiple clubs all wish to use the same route at similar times, to their respective fixtures. This can cause conflict between fans and strain policing and transport services. However, transport constraints were not considered in this model because this information is not readily available. An assignment problem could be used to assign fixtures to dates and timeslots in order to resolve pinch point concerns. For example, fixtures can be spread across Saturday and Sunday (dates) with different kick-off times (timeslots).

- **Superimposed slots with dates and kick-off assignment**

This model only concerns the development of a matrix of fixtures (schedule) with teams as rows and slots as columns. These slots are not assigned any specific dates for when they should take place. It is therefore assumed that each slot will be assigned to match-weekends by the league authorities. To maximise TV viewership, the TV broadcasters will then assign which teams will play on Saturday and Sunday as well as the kick-off times of each fixture in a slot.

A final note concerning scope and assumptions is that it is assumed that the solution schedule will be accepted by all stakeholders. This includes the clubs, police, Premier League and the Fixtures Working Party. As a result, the solution schedule can be compared to the official published schedules.

6.2 EPL Scheduling Model

6.2.1 Solving methodology

The solving methodology used to formulate the model is presented in Figure 6. The methodology consists of three separate engines which are used in sequence. Matlab R2013b was used to code and run the separate engines (see Appendix G).

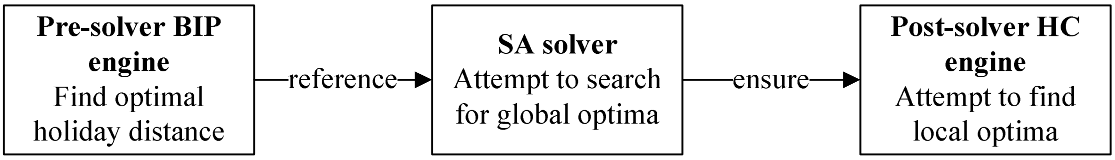

Figure 6: Solving Methodology

First the BIP (binary integer program) engine is used to determine the *reference value* for the holiday period sub-problem required by the SA (simulated annealing) solver. The SA solver is then used to find a solution of high quality. Lastly, the HC (hill-climbing) engine is used to *ensure* that the solution from the SA solver is the local optima of the region.

6.2.2 Pre-solver BIP engine for holiday period travelling distances

The SA solver requires certain initialisations which are discussed in the next section (Section 6.2.3). One such initialisation is the optimal travelling distance over the Boxing Day and New Year's Day fixtures. This is used as a reference point for constraint C107 in the SA solver. The optimal distance was found using a binary integer program.

- **Objective and constraints formulation**

The objective is to minimise the distance travelled by visiting teams (i.e. distance between opposition stadiums). The problem is subject to: no local derbies (C103), no strength group fixtures (C102b), there must be a break for all teams (C12c) and reverse fixtures are prohibited (C108). A complete mathematical formulation is presented in Appendix D.

- **Modelling**

The mathematical model was translated into Matlab code. A user may use Matlab's built-in *bintprog* tool but due to the size of the problem (800 variables) this can consume hours of processing time. CPLEX (CPLEX_Studio1251) Matlab plug-in is an alternative and was used instead. This used a tenth of a second to solve the problem.

The optimal distance found by the BIP is then input into the SA solver.

6.2.3 SA solvers

This section discusses the two different simulated annealing strategies used to solve the EPL scheduling problem. For convenience (i.e. easier to discuss, refer to and remember), they have been called the 'Da Vinci solver' and the 'Tesla solver'.

The objective and constraints for the EPL scheduling problem are discussed, the high-level simulated annealing framework is laid-out, the simulated annealing parameters are defined, the initial construction method is presented and the neighbourhood strategy is described.

- **Objective and constraints formulation**

The objective, as defined in Section 5.2, is to minimise the accumulated penalties of constraint violations with regards to candidate EPL schedules. All the constraints and constraint weights, defined in Section 5.3.1 and Section 5.3.2 respectively, are considered in the algorithm. The mathematical representation of penalty assignment can be seen in Appendix E. Problem parameters for the 2013/2014 EPL are shown in Section 5.4.

- **High-level code framework**

The pseudo-Matlab frameworks of the aforementioned solvers are shown at a high level:

Framework

```
BEGIN  Simulated Annealing framework
       ----------------------------------------------------------------------------------------------------------------
       \INITIALISE

       Define parameters and initialisations;
       Initial Trial Solution = Call Initial Trial Solution function;
       Current Candidate = Initial Trial Solution;
       ----------------------------------------------------------------------------------------------------------------
       SOLVER (insert Da Vinci or Tesla solver here)
       ----------------------------------------------------------------------------------------------------------------
       FUNCTIONS

       Initial Trial Solution function; (Description: Use polygon method to construct an initial
       feasible trial candidate solution)

       Neighbourhood Switch function; (Description: Use a neighbourhood search strategy to select
       an immediate neighbour of the current candidate sent to this function)

       Neighbourhood Acceptance function; (Description: Use the probability of acceptance function
       to decide whether to accept the trial candidate solution)

       Find Best Candidate from Population function; (Description: Find the candidate with the
       lowest total cost from the population)
       ----------------------------------------------------------------------------------------------------------------
END    Simulated Annealing framework
```

Da Vinci solver

\SOLVER (DA VINCI)

FOR LOOP temperature: $T_1 \rightarrow T_{temperature\ schedule}$

FOR LOOP counter: $1 \rightarrow \beta_{epoch\ length}$

Neighbourhood Candidate = Call Neighbourhood Switch function (send Current Candidate);

Current Candidate = Call Neighbourhood Acceptance function (send Current Candidate and Neighbourhood Candidate);

END counter loop

END temperature loop

Best Candidate = Call Find Best Candidate from Population function;

In the Da Vinci solver presented above, a predetermined number of iterations for each temperature in the temperature schedule are performed. There is a fixed number of temperatures ($T_{temperature\ schedule}$) and a fixed number of iterations ($\beta_{epoch\ length}$).

Tesla solver

\SOLVER (TESLA)

Best Candidate = Current Candidate;

WHILE LOOP temperature > T_{min}

counter = counter + 1;

Neighbourhood Candidate = Call Neighbourhood Switch function (send Current Candidate);

Current Candidate = Call Neighbourhood Acceptance function (send Current Candidate and Neighbourhood Candidate);

IF Current Candidate < Best Candidate

counter = 0;

Best Candidate = Current Candidate;

END if statement

IF counter = $\beta_{non\text{-}improving\ epoch\ length}$

counter = 0;

temperature = temperature.α

END if statement

END temperature loop

In the Tesla solver above, the temperature decreases at the predefined cooling rate (α) only once a maximum number of non-improving moves have been tested ($\beta_{non\text{-}improving\ epoch\ length}$). This solver stops once a predefined stopping temperature is reached (T_{min}).

- **Simulated annealing parameters**

Setting parameters is an important aspect of random search techniques like simulated annealing. If poorly set the performance in terms of solution quality and reliability may be significantly hindered. Solution quality here concerns finding the solution with the lowest total cost, and reliability concerns finding similar solutions repeatedly.

Parameter settings used in this solver were based on trial and error and suggestions from the literature. A combination of parameter setting methods from Hillier and Lieberman [41], Park and Kimto [52] and Anagnostopoulos et al. [44] were considered. Hillier and Lieberman [41] presents a simple parameter setting method but does not justify this method. Park and Kimto [52] is particularly useful as it identifies parameter setting methods in the literature and presents a systematic procedure to determine parameter values. Anagnostopoulos et al. [44] sets the parameter values of their simulated annealing solver, used to schedule the Major League Baseball TTP problem in the United States, based on experimental designs.

The solver parameters used in the Da Vinci and Tesla solvers are discussed next:

Acceptance function

A new candidate solution is accepted if a uniform random number is less than the probability of acceptance function.

$$\text{Prob(acceptance)} = e^{(Z_c - Z_n)/T}$$

Where Z_c is the total cost of the current candidate and Z_n is the total cost of the next neighbourhood candidate. For minimisation problems, if $Z_c\text{-}Z_n \geq 0$, the neighbourhood candidate is accepted as the new current candidate solution. This is called a *downhill move* (decreases total cost) or a *sideways move* (no improvement in cost). Conversely, if $Z_c\text{-}Z_n < 0$, the neighbourhood candidate is accepted as the new candidate with the probability function described above. Here, T is a reducing control parameter called *temperature*. Neighbourhood candidate solutions accepted this way are called *uphill moves* (increases total cost). Uphill moves are essential to simulated annealing as it allows the solver to escape local minima in search for the global minima.

Temperature schedule and epoch length

Temperature T is the control parameter used in the probability of acceptance function. Temperature governs the willingness of the simulated annealing solver to be random in its search for the global optima. At high temperatures, the search can be very random in a sense that the probability of acceptance of a non-improving neighbourhood candidate is high. As temperature decreases, the search randomness decreases since the probability of acceptance decreases. High temperatures aim to identify, through diversification, the promising regions which may contain the global optima while lower temperatures focus on finding the local optima, through intensification, of the promising regions.

There are three aspects concerning temperature. They are the initial temperature, cooling rate and epoch length.

Initial temperature T_1

Park and Kimto [52] notes that if the initial temperature is too high it may cause poor performance. However, according to Kirkpatrick et al. (1983), as cited by Park and Kimto [52], the temperature must be high enough to make the initial probability of accepting candidates to be close to one. Hillier and Lieberman [41] suggest using an initial temperature of $T_0 = 0.2Z_0$ but assumes that the initial trial solution is of good quality.

- Da Vinci solver: Considering the poor quality of the initial trial solution used (polygon construction discussed later in this section) and after experiments, it was found that the following produced the best results.

$$T_0 = 0.01Z_0$$

To initialise the solver, the total cost of the first candidate is $Z_c = Z_0$ where Z_0 is the total cost of the initial trial solution. For reheats, discussed later, Z_0 would be the associated cost of the best solution found from the previous reheat.

- Tesla solver: The initial temperature in this solver is based on the fast cooling strategy devised by Anagnostopoulos et al. [44]. An initial temperature of $T_0 = 700$ is used in their case. However, for the EPL scheduling instance, it was found that using a higher temperature produced improved solution quality.

$$T_0 = 900$$

Temperature schedule

Temperature is commonly reduced using a proportional function [52]. This proportional cooling function is represented as follows:

$$T_k = \alpha T_{k-1}$$

Here, the k^{th} temperature in the temperature schedule is a function of the previous temperature T_{k-1} and the cooling rate α, where $(0<\alpha<1)$. A cooling rate closer to zero would rapidly decrease the temperature which could lead to poor solution quality. Conversely, a cooling rate closer to one gradually reduces the temperature which could result in higher solution quality but at the expense of speed.

Park and Kimto [52] presents different cooling functions and methods used to obtain cooling rates. Hillier and Lieberman [41] suggest a cooling rate of $\alpha = 0.50$, and a temperature schedule of five temperatures. A common method used in the literature is to set the cooling rate based on a desirable probability of acceptance at low temperatures.

- Da Vinci solver: Here, the cooling rate used $\alpha = 0.80$, is slower than that proposed by Hillier and Lieberman [41]. It was found that since the cooling rate was slower, a larger temperature schedule was required. A temperature schedule consisting of ten temperatures was found to be adequate.

$$T_1 = T_0 = 0.01Z_0,$$

$$T_2 = 0.8T_1,$$

$$T_3 = 0.8T_2,$$

$$\dots$$

$$T_{10} = 0.8T_9$$

With this temperature schedule, a balance between diversification and intensification is achieved (i.e. a global search at high temperatures and local search at low temperatures). High temperatures allow uphill moves but as $T \rightarrow 0$, the simulated annealing solver moves towards a greedier search similar to hill-climbing.

- Tesla solver: Anagnostopoulos et al. [44] use a cooling rate of $\alpha = 0.98$ in their fast cooling strategy. Also, the temperature schedule is governed differently. In this case, the temperature is changed after a fixed number of consecutive non-improving moves are executed. For the Tesla solver, it was found that a cooling rate of $\alpha = 0.95$ produced good results in a reasonable time.

Epoch length

There are numerous ways one could set the number of iterations performed at each temperature (called 'epoch length'). For example, one could change the temperature each time a fixed number of consecutive non-improving moves have been executed, as done by Anagnostopoulos et al. [44]. Another option is to set it proportionally to the size of a problem instance or proportionally to the search space if it is known [52]. Park and Kim [52] set the epoch length proportional to the number of neighbourhood solutions of a given solution.

- Da Vinci solver: Park and Kim's' [52] concept was considered in setting the epoch length in the Da Vinci solver. The epoch length was set by multiplying the number of local neighbourhood moves by five.

In the neighbourhood search section presented in Section 3.3.5, it was noted that for $n = 20$ teams, there are $n(n-1)/2 = 190$ possible moves for each of the SwapHomes and SwapTeams neighbourhood strategies. If one uses the same logic, for $|s| = 38$ slots in a 2RR, the SwapRounds strategy would have $|s|(|s|-1)/2 = 703$ possible neighbourhood moves. Since all three strategies are used, this would amount to $190+190+703 = 1083$ possible neighbourhood moves. Multiplying this by a factor of five equates to approximately *5500*. Considering this, a good epoch length which yielded good results was approximated as that shown.

$$\beta_{\text{epoch length}} = 5000$$

- Tesla solver: Here, the same epoch length parameter was used. However, with the Tesla solver, the temperature only changes after consecutive non-improving moves.

$$\beta_{\text{non-improving epoch length}} = 5000.$$

Stopping Condition

- Da Vinci solver: The Da Vinci solver terminates after $\beta_{\text{epoch length}}$ iterations are performed at each of the ten temperatures in the temperature schedule. This is the stopping rule for the Da Vinci solver. The best candidate solution from the population is then selected.

- Tesla solver: This solver terminates once the temperature reaches a small target. Iterations beyond this temperature were found to be needless since no further improvements occurred.

$$T_{min} = 20$$

- **Concept of Reheats**

At low temperatures, the simulated annealing solver can get trapped in local optima because of the low probability of accepting uphill moves. Reheating is the idea of increasing the temperature to escape local optima at low temperatures, thereby increasing diversification. It is therefore a useful tool which can increase the solution quality.

The reheat strategy proposed by Anagnostopoulos et al. [44] is to reheat the temperature schedule after a fixed number of non-improving moves. The temperature is reheated to double the temperature at which the best solution was found.

A reheat strategy was judged to be needless for the Tesla solver because it appeared not to improve the solution quality. However, a reheat strategy devised for the Da Vinci solver appeared to improve the solution quality.

In the Da Vinci solver, a reheat was initiated after a temperature schedule had been completed. The best solution found so far was then used as the initial trial solution for the next run. A new temperature schedule was created using the same method discussed previously. However, this temperature schedule is multiplied by a factor which was found to consistently produce good results.

$$Z_{0\ new} = \text{penalty of best solution so far}$$

$$R_{increasing\ factor} = 2.5$$

$$T_{temperature\ schedule\ new} = R_{increasing\ factor} \times (T_{temperature\ schedule\ with\ new\ Z_0})$$

It was found that two reheats in the Da Vinci solver were ideal. After two reheats the solution generally converged to the same solutions. This suggests that the best hill from a particular region had been located and the local optima found.

$$R_{size} = 2$$

The pseudo-code below presents the modification (*in italics font*) to the simulated annealing framework, shown previously, required to incorporate reheats.

```
BEGIN  Simulated Annealing framework
       ⋮
       ------------------------------------------------------------------------------------------
       \SOLVER (modification)

       FOR LOOP reheat: R_1→R_size

              \SOLVER (insert Da Vinci or Tesla solver here)

              Initial Trial Solution = Best Candidate;
              Set Temperature Schedule (send Initial Trial Solution)
              Current Candidate = Best Candidate;

       END reheat loop
       ------------------------------------------------------------------------------------------
       ⋮
END    Simulated Annealing framework
```

If reheats are used, the program terminates after all reheats have been performed. This then forms the new stopping criteria.

- **Initialisation – the construction of a feasible initial trial solution**

In a basic simulated annealing solver, an initial candidate solution is required. Generally, it is advantageous to begin with one which appears to be good.

Due to the complexity of scheduling, it becomes increasingly difficult to create manual initial trial solutions for larger instances (the number of teams). Therefore, the polygon graph construction method described in Section 3.3.7 was used to create the initial trial solution. For the instance of $n = 20$ teams, the polygon construction output is presented in Appendix F.

- **Neighbourhood Search**

The neighbourhood heuristic strategy used comprised of a combination of three local search techniques found in the literature (SwapTeams, SwapRounds and SwapHomes). The simulated annealing solver switched between each strategy after a fixed number of iterations (called the 'changeover length'). It was found that smaller changeover lengths performed better than lengthy ones. A good changeover was found by trial and error to alternate in tandem.

c/o = 1

This way the disparity between the usage of each strategy at each temperature is minimised. As a result, the search quality of the local neighbourhood increases. It is important to note that due to the nature of the neighbourhood techniques used, only neighbours corresponding to feasible schedules are visited.

As stated in the neighbourhood section of Section 3.3.5, each technique used in isolation could result in local optimal solutions since the search area is inadequate. Therefore by using these three techniques together, the search area increases. Partial moves described by Anagnostopoulos et al. [44] and Ribeiro and Urrutia [46] may increase the search area further, but this involves ejection chains which would be too complex to code in the research timeframe.

6.2.4 Post-solver HC engine to increase solution quality

- **Integration rationale**

The basic idea is that the simulated annealing solver should at least identify the promising search regions within the search space. It may have identified the local minimum within this region or it may have not. After the simulated annealing solver, one could use the hill-climber engine to search for the minima in the immediate search vicinity. Usually, if the local minimum has already been reached by the simulated annealing solver, the objective value (total cost) will gravitate back to this solution after diverging away from it. This confirms that the local minimum has been found.

- **Hill-climbing engine formulation**

As mentioned in the literature survey, a hill-climber is a local search procedure. There are many derivatives of hill-climbing but they all aim to make incremental improvements to a solution until it can no longer be improved. Hillier and Lieberman [41] describe a *steepest-hill, mildest-descent* strategy. Here, all immediate neighbours of a solution are tested and the one which provides the best total cost is accepted (called the 'steepest-hill'). If there are no improving moves, the neighbour which weakens the total cost the least is accepted (called the 'mildest-descent'). This process occurs for a predefined number of iterations.

A simple hill-climbing engine comparable with the steepest-hill, mildest-descent is used in this research. In this case any improving move is accepted while any non-improving move within a percentage from the current candidate's total cost, defined as the 'hill-climbing factor', is accepted. This may appear similar to the simulated annealing methodology at low temperatures but at low temperatures, the simulated annealing solver only accepts non-improving moves with a small probability.

A hill-climbing factor of 3% appeared to produce the best results. It was found that assigning a larger factor often resulted in sharp uphill moves away from the local minimum and in some cases to adjacent hills with weaker solutions.

$$HC_{factor} = 1.03$$

A sufficient number of hill-climbing iterations were found by trial and error. It was observed that no further improvement in the total cost occurred for larger lengths.

$$HC_{length} = 1000$$

- **Greedy-search heuristic alternative**

Alternatively, as an aggressive greedy-search procedure, the hill-climber can be solely employed. This procedure would rapidly find solutions with smaller total cost but at the expense of solution quality. This is because it selects improving downhill moves while only selecting mild uphill moves on the way to a solution. Therefore, diversification across the search space is minimised and sacrificed for intensification and speed. As a result, generally local optima, rather than good solutions, are found.

6.2.5 Summary of SA strategies

A summary of the solver parameters used in the Da Vinci and Tesla simulated annealing strategies are presented in Table 10 and Table 11.

Table 10: Simulated Annealing solver parameters

	$T_{schedule}$	$T_{schedule\ size}$	T_0	α	$\beta_{epoch\ length}$
Da Vinci	$T_k = \alpha T_{k-1}$	k = 1,2,..,10	$0.01Z_0$	0.80	5000
Tesla	$T_k = \alpha T_{k-1}$	$T_k > 20$	900	0.95	5000 (non-improving)

Table 11: Simulated Annealing solver add-on parameters

	Neighbourhood strategy order	c/o	R_{size}	$R_{increasing\ factor}$	HC_{length}	HC_{factor}
Da Vinci	Swap Teams, SwapRounds, SwapHomes	1	2	2.5	1000	1.03
Tesla	Swap Teams, SwapRounds, SwapHomes	1	-	-	1000	1.03

Flexibility

The model methodology consisting of the pre-solver BIP engine, the SA solvers and the post-solver HC engine was coded into Matlab. The Matlab files are included in the electronic appendix (see Appendix G). The code was constructed and coded in a manner which allows for flexibility. A user may access the code and under the solver parameters section, change these parameter settings. For example, a user may change the number of reheats in the SA solver and decide where, and whether, to use a post-solver hill-climber. All the parameters shown in the above two tables can be changed.

It must be noted that the algorithm settings and solver parameters used in the Da Vinci and Tesla solvers appeared to produce the best performance from many experiments. Other strategies can be tested by the sports scheduling world to suit each problem's needs.

6.3 Verification

The model was verified to ensure that it was logically and mathematically sound. Logic verification incorporates the depiction of the EPL sport scheduling problem into a mathematical model. Mathematical verification on the other hand entails the verification of the mathematical model as well as the model output.

Systematically following the methodology outlined in Section 4 and translating the problem into a simulated annealing mathematical model demonstrates a logical approach. Supervisor revision ensured logical rationality.

Mathematical verification was attained on a number of levels. At the code level, functions were programmed separately and tested for robustness by trying extreme values and testing for accuracy of output against expected output. In this way, bugs were eradicated. An Excel verification file is included in the electronic appendix (see Appendix I). The verification process was as follows:

- Verified the Neighbourhood Switch function by observing the program changes to test candidates and comparing the observed output to the expected output.
- Verified the *HAP Set* penalty function by observing the penalties assigned to test sets and then comparing them to expected penalties for that test set.
- Verified the *T Set* penalty function the same way as the *HAP Set* penalty function.
- Verified the Boxing Day and New Year's Day total travelling distance by comparing the observed distance for assigned fixtures to the expected distance for those fixtures.

Once the functions were assembled into a holistic system, the model was tested at a systems level. Here, the penalties assigned to the best solution identified from the population were verified using an Excel verification framework (see Appendix I). Compilation bugs causing deviations were rectified.

A Matlab data point graph was coded to graphically illustrate the total costs of the candidates in the population and to also verify the identification of the candidate from the population by highlighting the best solution data point (i.e. lowest total cost). Lastly, the solution was tested manually through Excel logic to ensure feasibility (see Appendix I).

This logical and mathematical verification process ensured that all bugs were identified and rectified.

7 OBSERVATIONS

In this section, the data gathered from the output of the two strategies developed in Section 6 are presented. The problem parameters used to gather this data are those of the 2013/14 EPL season. Regarding hardware and software specifications, an Intel Xeon quad-core CPU @ 3.10GHz (model E31225) was used to run the program coded in Matlab R2013b (64-bit) on a Windows 7 operating system.

7.1 Trial Neighbourhood Observations

To determine the solution quality of the two strategies over time, every trial solution (i.e. candidate) made by each solver was recorded. All improving neighbourhood moves made by the two solvers over time are depicted as data points which are plotted in Figure 7.

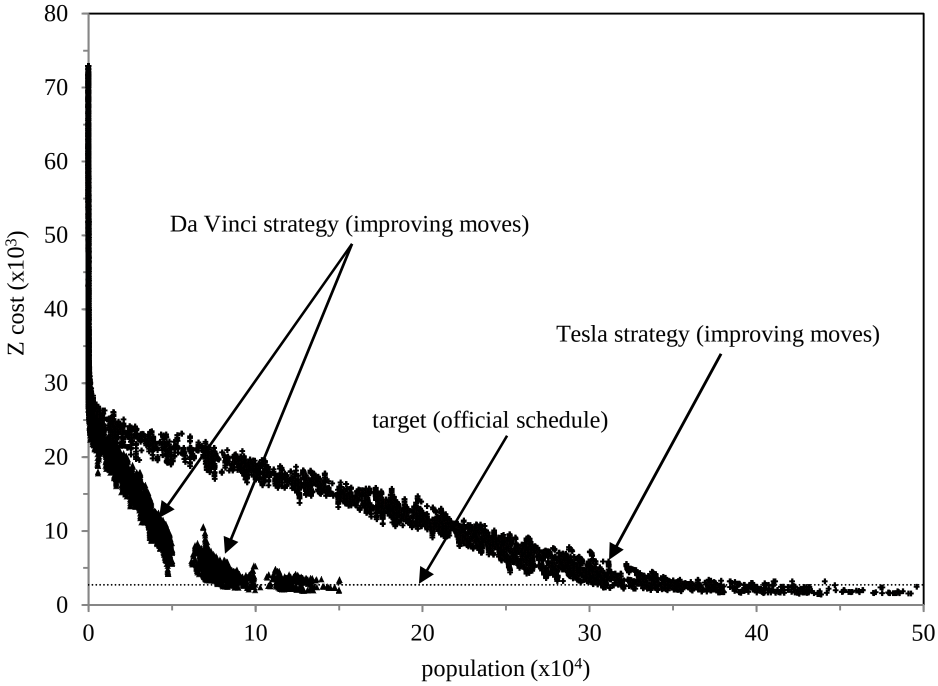

Figure 7: Observations of the Tesla and Da Vinci strategies (30 runs)

The x-axis represents the population of solutions tested and the y-axis represents the total cost of the best solution found at the current population size[6]. A visual reference line is inserted into the figure. Since no benchmark exists for the EPL scheduling problem, this reference line is the total cost associated with the official schedule (Z cost of 2727).

[6] Note that the axis values must be multiplied by the metric represented in the axis labels. For example, the x-axis population figure of 50 must be multiplied by 10^4 amounting to $500x10^3$ while the y-axis cost of 70 must be multiplied by 10^3 which amounts to $70x10^3$.

The Central Limit Theorem states that regardless of the population distribution, if the sample size is sufficiently large (i.e. N ≥ 30), the sample mean distribution tends to be normal and equivalent to the mean of the parent distribution. Therefore, in line with the Central Limit Theorem, a sample size of 30 runs for each strategy was used to collect the data.

7.2 Sample Runs

A sample run observed for each strategy is presented in this subsection. Figure 8 and Figure 9 is a sample run observed from the Tesla strategy while Figure 10 and Figure 11 is a sample run from the Da Vinci strategy. It was observed that all runs created similar trends for the respective strategies.

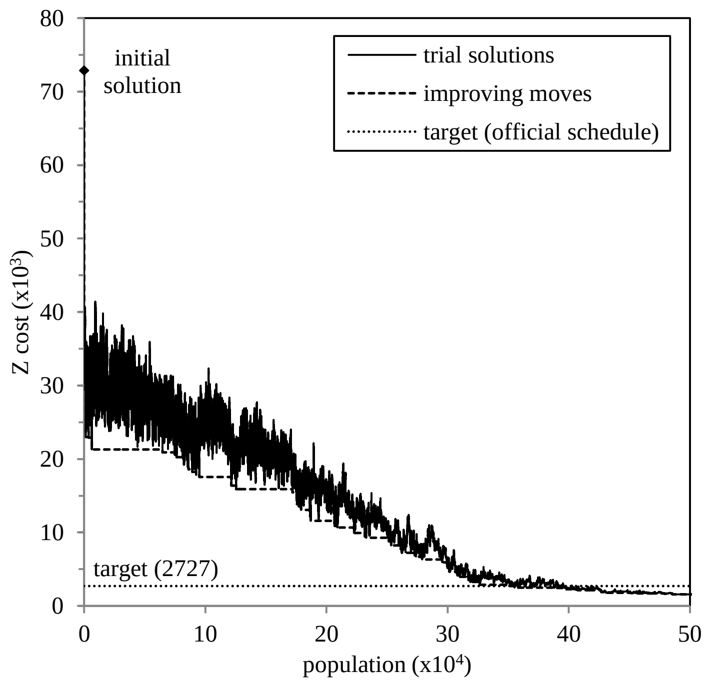

Figure 8: Tesla solver sample observation

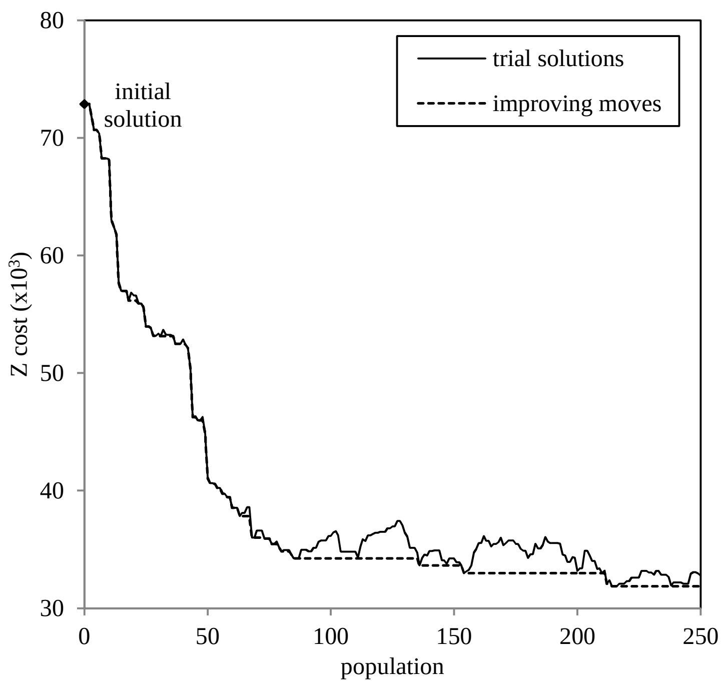

Figure 9: Tesla solver sample observation (first 250 moves)

In the sample run in Figure 8, all trial neighbourhood moves accepted by the simulated annealing acceptance function in the Tesla solver are represented with a solid line. Additionally, a dashed line representing the best solution found so far is shown. Since the initial moves result in a steep decline in cost, a data point indicating the initial solution is shown and labelled in this figure. To further illustrate the steepness of the initial moves observed, the first 250 moves of the sample run presented in Figure 8 are magnified and shown in Figure 9.

Likewise, Figure 10 and Figure 11 represent a sample run and the initial steepness observed from the Da Vinci solver.

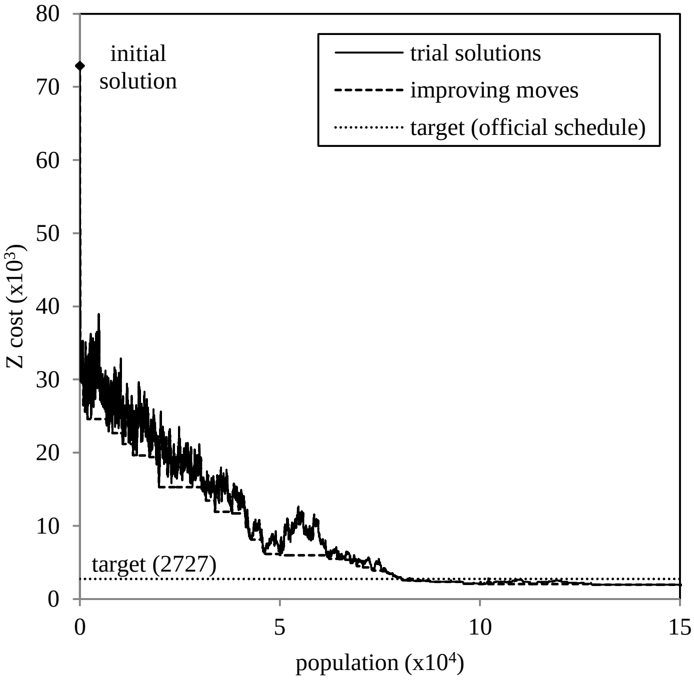

Figure 10: Da Vinci solver sample observation

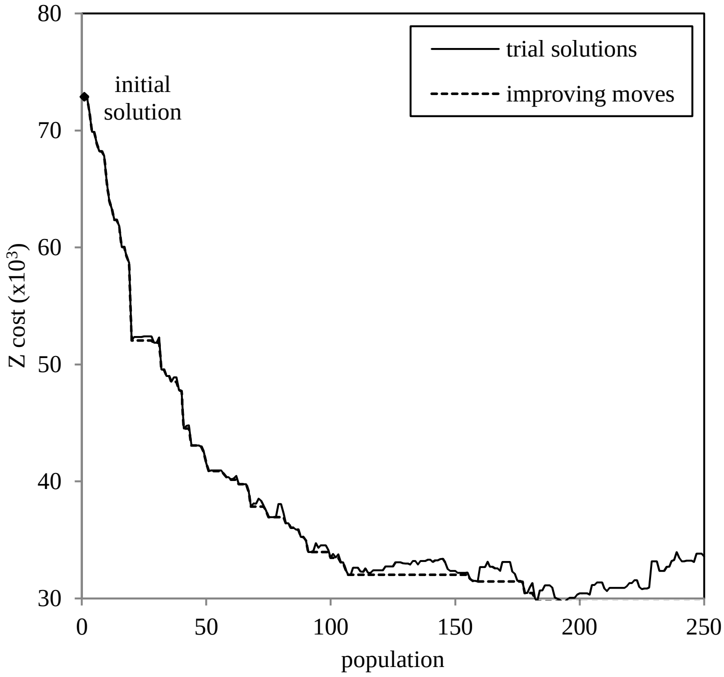

Figure 11: Da Vinci solver sample observation (first 250 moves)

8 RESULTS

8.1 Performance Indicators

A performance index is necessary to assess the performance of the two simulated annealing solvers. Three criteria in the performance index are used to assess the performance namely, solution quality, reliability and speed. Solution quality concerns the total cost incurred for a solution and the gap to a benchmark, reliability concerns the variation of solution quality over numerous runs and speed is concerned with the time to solve the problem. Solution quality is of considerably more importance than speed, while reliability is of less importance than solution quality.

8.2 Performance

The table below, Table 12, shows the performance of the two solvers in question. Here, outliers from the sample space were omitted from the calculations since they account for a small probability of a normally distributed sample mean. A sample space of N = 30 runs was observed and the arithmetic mean of the total costs Z($\bar{x}$) was calculated for each solver.

From this, the performance index was determined. The gap($\bar{x}$) is the arithmetic mean difference from the official schedule's total cost i.e. $100\left(\frac{average\ total\ cost\ of\ solutions\ found}{total\ cost\ of\ the\ official\ 2013/14\ schedule}\right) - 100$. A positive gap suggests that on average the solution quality of the solutions found were worse than the official schedule while the contrary is true for a negative gap. The variation of solution quality is represented by the standard deviation for the sample Z(s), while the speed is measured in time measurements of minutes.

Table 12: Results from sample population

	Sample space		Performance Index		
Solver	N	Z($\bar{x}$)	gap($\bar{x}$)	Z(s)	time($\bar{x}$)
Tesla solver	30	2025	-26%	367	40 min
Da Vinci solver	30	2843	+4%	493	12 min

Looking at the performance index, the Tesla solver outperforms the Da Vinci solver in the key criteria of solution quality and reliability. With regards to speed, the Da Vinci solver performed the best but this is of considerably less importance. It is important to note that although the Da Vinci solver is outperformed by the Tesla solver, its solution quality is comparable to the official schedule target since its gap percentage is close to zero.

8.3 Validation

The best solver selected, based on the performance index, was determined to be the Tesla solver. A simple validation process was conceived to determine the validity of the selected solver. Since the EPL scheduling problem is substantially different to other football scheduling instances, which all concern the break minimisation problem (see Section 3.4), the validity of the Tesla solver is compared to the official schedules of three EPL seasons. These include the 2013/14 EPL season, the 2012/13 EPL season and the 2011/12 EPL season. Problem parameters for these seasons can be seen in Appendix B, while the electronic appendix contains the Matlab problem parameter code for these seasons (see Appendix G).

Table 13 below presents the results of the proposed solutions found for each of these three instances, together with the official schedules. The proposed solution was selected from a sample of 25 runs (i.e. 30 runs less the outliers observed) since there is a small probability of obtaining outliers. Therefore, the performance index gap is based on the proposed solution's total cost while the time remains the arithmetic mean of the sample runs used. Soft constraint violations (hard constraints in parenthesis) of both official and proposed schedules (except for C107), along with the total distance travelled by all teams over the Boxing Day and New Year's Day fixtures (C107), are shown separately.

Table 13: Validity

	Official schedule			Proposed schedule			Performance Index	
Year	violations	distance	Z	violations	distance	Z	gap	time ($\bar{x}$)
2013/14	68	4120 km	2727	57	3130 km	1574	-42%	40 min
2012/13	59 (20 hard)	3066 km	23997	46	3084 km	1614	-93%	40 min
2011/12	61	3074 km	2538	40	2840 km	1534	-40%	40 min

It can be seen by the gap column in the performance index that the Tesla solver consistently produced better solutions than the official schedules published over the three seasons. It is important to note that the proposed solutions were not necessarily the best schedules found but rather those schedules that the author believes are likely to be accepted by the league owners and all relevant stakeholders. This is because in each of the best solutions found, constraint C103 (i.e. paired fixtures over the Boxing Day and New Year's Day fixtures) was violated exactly once (due to constraint penalty trade-offs) while the proposed schedules did not violate it. It is the responsibility of the Fixtures Working Party to judge if these schedules are acceptable or not. The abovementioned official, proposed and best schedules found along with a table of the total cost breakdown of the each schedule are presented in Appendix H.

9 DISCUSSION

9.1 Problem Solving Techniques and Solver Parameters

Disregarding the weights assigned to constraint violations, it was observed that the solution quality (i.e. total cost and gap to benchmark) was highly sensitive to two aspects. The first is the simulated annealing problem solving technique used and the second is the associated solver settings, or solver parameters. It was deduced from the literature that there is no standard way of creating a simulated annealing problem solving technique. Similarly, there is no single way of setting a metaheuristic's associated solver parameters. Rather, these two aspects are problem specific and found through trial and error.

Correspondingly, the Da Vinci and Tesla problem solving techniques were created, which are derivatives of simulated annealing, and based on techniques and solver parameter setting methods found in the literature. A range of solver parameters for each solving technique were subsequently tried and selected based on trial and error experimentation.

- **Initial trial solution**

An initial trial solution for the EPL is difficult to create manually, so it was created using the polygon construction method. This method has its roots in graph theory. The polygon method provides a quick and simple way of formulating an initial feasible schedule. As can be seen in Figure 8 and Figure 10 in Section 7.2, the total cost associated with this initial solution is large when compared to the target level. It is recommended in the literature that a good initial solution should be used. When using initial solutions of poor quality, the simulated annealing solver spends the majority of processing time searching for promising regions to explore in the search space. Looking at the aforementioned figures, it can be seen that only a small portion of the search is close to the target level and only occurs at the latter stages of the search. Attaining a good solution to be used as the initial trial solution proved to be difficult using manual manipulation. Hence, the solution produced by the polygon method had to be used as the initial solution.

- **Solver parameters**

The temperature schedule was found to be a solver parameter setting which influenced solution quality. Contrary to suggestions found in the literature concerning setting the initial temperature in a temperature schedule (Section 6.2.3), ideal initial temperatures were found by trial and error to be lower than those suggested. When using higher initial temperatures, it was found that the search decreased to the target level at a slower rate, often resulting in poor solutions. It appears that with the cooling rate set closer to one than zero (subsequently discussed), coupled with a fixed temperature schedule, the probability of accepting uphill moves (i.e. solutions with larger total cost) is prolonged in the search when using higher initial temperatures. As a result, a lack of hill-climbing occurs at the latter phases of the search (little intensification), an important facet of the simulated annealing process, since the probability of accepting uphill moves is still too high.

It was observed that using slower cooling rates, rates closer to one, in conjunction with a larger temperature schedule produced the best results. This is the main difference between the Da Vinci and Tesla solvers. The Tesla solver uses a slower cooling rate and a temperature schedule over seven times larger than that of the Da Vinci solver. Slower cooling rates with larger temperature schedules allow the search to sufficiently explore the hills in a local region before gradually selecting the best hills to search further. On the other hand it was found that faster cooling rates with confined temperature schedules often resulted in steeper declines in total cost over time but with inferior solution quality. It is likely that since the local hills are not sufficiently explored in this manner, inferior search regions are searched in the latter phases of the temperature schedule and cannot be escaped.

While the Da Vinci solver used a fixed temperature epoch length, the Tesla solver used a more flexible one which only reset once a fixed number of non-improving moves were found. It was found that that this allowed the Tesla solver to improve its search for promising search regions in the search space. The reason for this is twofold. First, it was observed that promising search regions were more often found at certain temperatures, possibly because such regions lie in a hilly area. Secondly, it was found that the Tesla solver was best at exploiting this. It is therefore deduced that since the Da Vinci solver has a fixed epoch length, if it does not find and explore the promising hills in this hilly region, which it often did not, the solution quality would be weak. Conversely, the Tesla solver was capable of finding and exploring these promising hills. For this reason, with regards to solution quality, the Tesla solver outperforms the Da Vinci solver.

9.2 Reheats and Hill-Climbing Add-ons

The Da Vinci solver employed supporting add-ons which allowed it to improve its solution quality. These add-ons include reheats and hill-climbing. In test runs, it was observed that without these add-ons, the Da Vinci solver would converge to weak solutions (i.e. solutions with a total cost above the target level). This can be attributed to the solving technique not effectively finding promising search regions to explore in the search space. Respectively, an inadequate temperature schedule (a function of cooling rate and initial temperature) and fixed epoch length is possibly causing the premature convergence and ineffective search for promising search regions.

Since it is difficult to find a set of simulated annealing solver parameters which consistently produce good results for the EPL scheduling problem, the concept of reheats and a hill-climbing add-on were introduced. It appeared that reheats were effective in preventing the Da Vinci solver from converging prematurely since it allowed the search to escape local optima found in the search space by allowing uphill moves away from current hills.

Additionally, it was found that employing a hill-climber at the end of the Da Vinci search slightly improved the solution quality. This suggests that although the Da Vinci solver, with the reheat add-on, is capable of finding promising hills in the search space to explore further, it is not effective in finding the optima in this vicinity. This is because with a low probability of accepting uphill moves in the latter phases of the simulated annealing search, no matter how small, any hill is unlikely to be climbed. It was found that using a hill-climber, with a more relaxed acceptance criteria, was effective in passing small hills to find optima in the immediate search vicinity.

Interestingly, contrary to the Da Vinci solver, it was found that no improvements in solution quality were found in the Tesla solver when using either of the aforementioned add-ons. This suggests that the solver technique and associated solver parameters were effective in finding promising regions to explore and furthermore, were subsequently capable of finding the local optima.

9.3 Neighbourhood Search

The neighbourhood search technique is a vital aspect of any metaheuristic. Often, the limits of a search technique can have a detrimental effect on the final solution quality of a solver. Two key points were identified regarding the neighbourhood search quality. The first concerns memory while the second concerns permutations.

- **Memory**

Due to the nature of simulated annealing, neighbourhood moves are based on a continuous random search of the local region. The acceptance of such a move is determined by the probability of acceptance function. However, a string of random moves which may lead to good results is not 'remembered' by simulated annealing solvers. Instead, these solvers may easily undo this opportunity with subsequent random moves. Therefore, the introduction of some memory, such as a Tabu list in the Tabu Search metaheuristic methodology, may improve the search methodology of the Tesla and Da Vinci solvers. For example, if a string of random moves results in a reduced total cost, then forbid the solvers from alternating these changes for a fixed number of moves. A simpler option may be to introduce a basic Tabu list which forbids recent changes for a fixed number of iterations. Memory can improve diversification and intensification by allowing the solver to select smarter moves at a quicker rate to more promising regions in the search space, rather than continuously making random moves.

- **Permutations**

The second aspect influencing the quality of the neighbourhood search strategy concerns the actual changes made to the schedule. SwapTeams, SwapRounds and SwapHomes are the three neighbourhood strategies used in combination in this research. None of these use ejection chains. If the scheduling problem only concerned a 1RR, these moves would sufficiently explore all possible permutations in the search space. However for a 2RR, as highlighted by Ribeiro and Urrutia [46], if ejection chains are not used, one may quickly get trapped at schedules governed by the structure of the polygon method. This means that, if one looks at the proposed schedules in Appendix H, for every slot of fixtures in a round, there is an equal and opposite slot of fixtures in the other round.

Ejection chains, which allow infeasible moves and then create feasible schedules, can break the structure of the initial polygon by modifying the one-factors to form a new one-factorisation set built by the polygon method. Although complicated to code, if ejection chain moves are used, a larger search space will exist and higher quality solutions will be found. Interestingly, by looking at the official schedules of the EPL in the same appendix, it can be seen that equal and opposite slots exist in either half of the 2RR and therefore it is deduced that the league creators also do not make use of ejection chains.

9.4 Trend Observations

Sample observations of the two simulated annealing problem solving techniques resulted in decreasing trends. A sample observation of each can be seen in Figure 8 and Figure 10 of Section 7.

In both cases, the trial solutions line produced a wedge shape with large oscillations at low population levels and smaller oscillations as the population level increases. Since the temperature variable in the probability of acceptance function at the start of the temperature schedule is high, the probability of accepting any uphill move is highly probable. This results in large oscillations between trial solutions at the low population levels. Conversely, as the temperature variable decreases, the probability of accepting uphill moves decreases. This results in the decreasing oscillation noise as the population increases. It can be seen that eventually almost negligible oscillations occur towards the end of the population until zero oscillation noise occurs. This suggests that a promising hill has been found by the solver and is being explored further until the local optimum, possibly global, is found and cannot be improved further.

If one observes the Da Vinci solver trend more closely (Figure 10), it can be seen that there are two disturbances, one at the 50×10^3 population level and another at the 100×10^3 population level. This is the stage where the solver employs reheats. It was found by experimentation that if reheats were not employed around these population levels the solver would converge to a solution before reaching the target level, as discussed previously. The first disturbance clearly shows the trial solutions diverging away from the decreasing trend before returning to a decreasing trend beyond the previous best solution, analogous to the solver escaping local hills and moving to more promising hills in the search region. The solver can escape these local hills because the temperature variable is increased in the probability of acceptance function at the point of reheat. It is important to note that, although less significant improvement occurs, the same happens at the second disturbance.

A straight line towards the end of the population in the Da Vinci solver sample observation can be seen (Figure 10). This is the region where the hill-climbing add-on is executed. Since the line is straight for the majority of the time, this indicates that no better hill exists in the search vicinity. A small improvement in cost is found at the end of the search which represents the local optimal solution. However, in some trials, more improvement in cost was observed during this phase of the solver's search. With regards to the Tesla solver which also used a hill-climber, on every occasion it was observed that a straight line exists at the end of the population, which confirms that the local optimal solution was found.

Deceptively, both solvers produce sample observations which appear to decrease almost linearly from a cost of approximately $40x10^3$. This is not the case. In fact, as can be seen in Figure 9 and Figure 11, a rapid descent in total cost exists in the initial search phase. Since the total cost assigned to the initial trial solution is high, regardless of the probability of acceptance, almost any change improves the schedule and is the reason for this rapid descent. As for the appearance of a linear trend from the $40x10^3$ total cost region, this can be attributed to the cooling rate used. Since this cooling rate is slow, close to one, the trend appears almost linear. It was observed that a steeper trend appeared for faster cooling rates of approximately $\alpha < 0.5$, however the solution quality decreased.

9.5 Results

The performance index in Table 12 (Section 8.2) shows that the Tesla solver outperforms the Da Vinci solver in the key criteria of solution quality ($gap(\bar{x})$) and reliability ($Z(s)$). Although the Tesla solver produces superior solutions, the solution quality of the Da Vinci solver is comparable with the official schedules.

It is likely that the Tesla solver has a higher quality of solution because it uses a slower cooling rate and a flexible temperature schedule. Unlike the Da Vinci solver, the Tesla solver persists in exploring all trial solutions in the immediate region at each temperature before deciding which hills to explore next and subsequently reduce the temperature variable.

The Da Vinci solver was able to arrive at solutions quicker than the Tesla solver ($time(\bar{x})$). Primarily because the Da Vinci solver explored a population of $15x10^4$ as opposed to a population of approximately $50x10^4$ explored by the Tesla solver. Interestingly, it was found that employing computing memory management and processing techniques led to drastic reductions, of up to nine times, in solving time. It is for this reason that solving speeds cannot always be directly compared against other solving techniques (unless the speeds are orders of magnitude different). Computing techniques employed here to reduce solving time included initialising variables, predefining variables, replacing duplicate functions within the simulated annealing loop with variables and reserving computing memory space in the initialisation phase rather than building structures. A programmer competent in lean code and memory management can almost certainly improve the speeds of the program coded in this research.

9.6 Practicality

To establish the practicality of the proposed Tesla solver, with regards to the real-world EPL scheduling, the assumptions need to be justified and the validity of the model determined.

- **Assumptions**

Transport, local services strain, policing constraints and weight assignment are discussed here.

Two fundamental assumptions made in the model are that the proposed schedules do not put excessive strain on either the so-called ‘pinch-point’ transport routes or the local services. Pinch-point routes are public transport routes which rival fans would likely use to attend a match. Excess strain on local services (i.e. medical and policing) arises when two home games occur in close geographical proximity.

Strain on local services is reduced through constraint C07, which aims to prevent paired teams from playing at home in the same slot. Additionally, since rival fans are usually local rivals, rival fans are supporters of paired teams. As a result, constraint C07 can also effectively reduce the likelihood of rival fans using pinch-point routes.

It is therefore deduced that the two assumptions made will be justified if constraint C07 has a low penalty count. If one observes the total cost breakdown table, Table H10, it can be seen that constraint C07 is violated only once in the proposed schedules of the three seasons considered. Over the same period, constraint C07 was violated twice in the official schedules. Therefore the respective assumptions are justified and the proposed schedules are competitive. In practice, often such issues are minimised by the league simply by assigning some fixtures to Saturday and others to Sunday.

Concerning EPL policing constraints, it was stated in the literature that there are over 90 policing constraints (Section 3.2). Since it was difficult to attain every constraint, it was assumed that the high priority constraints were identified and considered in this research. This assumption can only be justified if the important policing constraints considered in this model are approved by the British police.

If deemed necessary, additional constraints can be formulated and coded. When adding constraints it is important to note that weight assignment is relative. Therefore, constraints of less importance have weights of less magnitude. Although the total cost will increase if constraints with small weight are added, it is highly unlikely that it would have an effect on the current solution. Conversely, adding constraints of larger weight may result in solutions which drastically differ from the current solution. The reason for this is that constraints of larger weight will dominate since their weights significantly impact the total cost.

Interestingly, it was observed that if the magnitude of some soft constraint weights were too large, the solution quality decreased. This is because as the probability of accepting uphill moves decreases over time, the likelihood of accepting moves which violate these larger weight constraints decreases, even if such a move potentially leads to good solutions. Similarly, it was found that introducing some infeasibility by reducing the weights of some hard constraints, resulted in increased solution quality. This allows the solver to accept uphill moves on its search path to promising regions in the search space.

The reliability of the weights assigned to constraints identified in this model, presented in Section 5.3.2, is another important consideration which can influence solution quality. Although a formal process was used to assign relative weights to constraints, they are in fact subjective. It follows that if the relative weight priorities are not accepted by the stakeholders, the acceptability of a solution decreases. However, the weights are constants and can be adjusted in the problem parameter function in the code.

- **Validity**

As can be observed in Table 13 (Section 8.3), it appears that the Tesla solver produced superior results to the official schedule for the three seasons considered. The solution quality and speed are discussed.

Schedules with an improvement in solution quality of at least 40% were found. However, as discussed previously, weight assignment can be a contentious issue and a source of questionable results. Therefore, if one also observes the violation columns, assuming that the relative weight priorities assigned are similar to those used by the league, it can be seen that the Tesla solver also consistently found schedules which violated fewer soft constraints while violating no hard constraints. It is important to note that the solution quality gap of the proposed schedule for the 2012/13 season is considerably better than the official schedule. In that season, as can be seen in Table H2 and Table H10, the league published a schedule which violated the hard constraint C104, which concerns facing opposition twice in a single round of a 2RR.

Since the solver employed for the three seasons is the same, the average solving time is expected to be the same. An average solving time of 40 minutes was observed for all three seasons. To place this in context, it was found in the literature (Section 3.2) that Atos, who have been scheduling the EPL league for almost 30 years, currently have an estimated solving time of up to 30 minutes. However, the solving technique, degree of lean code and computer specifications used by Atos is not known. Therefore, the solving time of the Tesla solver in its current state is comparable.

9.7 Novelty

Novelty is discussed here with regards to the originality and uniqueness of the EPL scheduling problem and the Tesla solver. The EPL scheduling problem is not a new problem but to date, only one article by Kendall [4] on this problem exists in the literature. Even so, discussed in Section 3.3.6, the article only considers the holiday period fixtures (i.e. Boxing Day and New Year's Day fixtures). It follows that although benchmarks exist for real-world problems, a benchmark does not currently exist for the EPL case. Considering that the EPL is the most popular and highest valued league in the world, more attention ought to be given to its schedules.

Extensive research which exists in the field of sports scheduling concerns the break minimisation and TTP distance minimisation problems. However, the EPL sports scheduling problem is a distinct problem (see Section 3.4). The EPL can be described as a highly constrained problem where single breaks are allowed, but not over the holiday period fixtures, where travel must also be minimised. This makes the problem difficult to solve using techniques traditionally used in sports scheduling such as integer programming, constraint programming and combinatorial optimisation.

Equally, the simulated annealing Tesla solver created is an original problem solving technique. Currently, in all published sport scheduling research completed since 1968, metaheuristic solving techniques are outnumbered by other solving techniques. Only about 30 of the 250 plus articles referenced in the online library published by Sigrid Knust [29] use metaheuristic techniques. This is because early research was dominated by graph theory and this was followed by a rise in integer programming techniques. However, it is observed that in the past ten years, metaheuristic solving techniques are steadily rising and beginning to dominated new research in this field. It is suspected that as time has passed, computing power has increased exponentially and the potential of metaheuristic techniques are being realised.

To compare the ability of the Tesla solver to what is currently used, one should compare the current methodology employed by the league and its results.

Reading literature in news articles on how Atos schedules the EPL league, it is predicted that Atos use a derivative of the first-break, then-schedule problem solving methodology. Kendall [4] is currently working with the league with regards to minimising distance over the two holiday period slots.

The solution quality and speed of the Tesla solver were discussed previously under the validity point in Section 9.6. The holiday period distance aspect of the solver in Table 13 is discussed here. Summing the distance columns in Table 13, it can be seen that the total distance travelled over the three seasons for the proposed schedule is 13% less than the sum total of the official schedules (official schedule sum total of 10260 km and proposed schedule sum total of 9054 km). Therefore, the Tesla solver is capable of finding high quality solutions which not only violates fewer constraints than the official schedules but also reduces the travel distance over the two holiday period slots. It must be noted that the scope in this research only considered the EPL and does not include the lower divisions which the league does consider. However, the EPL teams have larger fan bases and larger stadiums than lower divisions. Therefore, by minimising the EPL travel over the two holiday slots, less total travel by all fans should result. Therefore, it is suggested that lower division fixtures are built around the EPL schedule fixtures over these two slots.

Apart from the distance minimisation aspect of the EPL, except for the 2011/12 proposed schedule with a single violation, the Tesla solver was capable of finding schedules with semi-complementary *HAP Set* vectors for all paired teams (i.e. see C07 in Table H10). The official schedule published by the league violated this constraint twice in the 2011/12 season. Therefore, even though it is suspected that Atos uses a first-break, then-schedule problem solving technique which aims to prevent such pattern issues, the Tesla solver achieves slightly better but comparable results.

For the reasons discussed above, it appears that the Tesla solver is a capable and superior solving technique when used to schedule the EPL. It is capable of achieving both break scheduling requirements and distance minimisation requirements while minimising other constraint violations. Furthermore, it can achieve this without using combinatorial solving techniques as used by the league. Considering that there are numerous meetings with different stakeholders to discuss issues which may require a change to the schedule, metaheuristics can be ideal techniques to use. This is because the current schedule in question can be input as the initial trial solution of the search. Weights can be adjusted and new constraints can be added accordingly. This process can be repeated until a high quality schedule which addresses all stakeholder issues is found.

9.8 Transferability

Transferability concerns the ability of the Da Vinci and Tesla solvers in solving different types of optimisation problems (not necessarily sports scheduling problems). The nature of simulated annealing renders it easily transferable to solving any type of problem, from linear to non-linear problems. Simulated annealing becomes particularly useful in solving complex problems such as highly constrained and NP-hard problems which would otherwise require large processing power and time to solve using traditional techniques such as integer programming.

Selecting the solver, setting the solver parameters and formulating the neighbourhood strategy ultimately influences the solution quality of any problem and are necessary aspects of any metaheuristic solving technique.

Like any problem, a researcher must first formulate the problem. A suitable neighbourhood strategy, capable of searching the entire search space, or at least an acceptable portion thereof, must be also devised.

A researcher must then decide which of the two simulated annealing frameworks presented in this research bests suits the needs of the problem. Solver parameters must also be manipulated by experimentation to find a combination which most effectively solves the problem. Although the Tesla solver resulted in superior solution quality compared to the Da Vinci solver in solving the EPL scheduling problem, it will not necessarily be the case for other types of optimisation problems. Therefore both solvers should be considered before selecting one to pursue further.

Concerning the EPL problem presented in Section 5, the manner in which the problem is laid out enables for the adjustment of the problem's parameters (Section 5.4) and constraints (Section 5.3) according to a stakeholder's needs. Other sports scheduling problems, for which literature does not exist as yet, can also be solved using the core Da Vinci and Tesla simulated annealing solvers and neighbourhood search strategy presented in Section 6.2.3. The formulation and set of constraints specific to those problems would have to be developed.

This discussion is concluded with an illustration of the diverse sports scheduling capabilities of simulated annealing. It was found in the literature that this scheduling technique is commonly used to solve artificial TTP problems concerning baseball and ice hockey leagues. In addition, it was also found that this problem solving technique has been used in practice on two occasions. Once to schedule the Australian 1992/93 domestic cricket league [53] and on another occasion to schedule the national New Zealand basketball season of 2004 [54].

10 CONCLUSION AND RECOMMENDATIONS

10.1 EPL Scheduling Problem

Section 3.4 states that football sports scheduling problems and benchmarks exist for the Austrian, Chilean, Belgium, German and Brazilian leagues. Surprisingly, in the benchmark article presented by Nurmi et al. [3], a problem framework and benchmark does not exist for the EPL, which is the league with the largest following and value in the world. Nurmi et al. [3] subsequently challenges the sports scheduling community to solve the unsolved real-world instances such as this research.

It was highlighted in the same section that the nature of the EPL scheduling problem is different to other real-world football sports scheduling problems. This is because all those leagues with a benchmark concern either the break minimisation problem or the travelling tournament problem. In comparison, the EPL relaxes the break constraint and requires the travel over the holiday period slots to be minimised.

10.2 Objective Conclusions

Here, the answers to the three objectives posed at the beginning of the research, Section 2, are concluded.

- **Identify EPL scheduling stakeholders and their respective requirements**

Stakeholders involved in EPL scheduling were identified from the literature (Section 3.2). Key stakeholders were found to be the league authorities (i.e. the Premier League), the clubs, the fans, the police and the public transport authorities. Except for the latter, beyond the scope of this research undertaking, those requirements which could be found from the literature were gathered.

A formal weight assignment method was used to then assign priorities to stakeholder requirements. Apart from those requirements which must be met (i.e. hard constraints), relative weights were assigned to constraints (i.e. soft constraints) according to the estimated value of the EPL, UCL, UEL and FAC. Since the potential earnings from the FLC were determined to be comparatively small, it was not considered further.

Discussed in Section 9.6, the solution quality of this research is highly dependent on whether the significant stakeholder requirements were identified and whether the relative weight assignments used accurately depicts the stakeholder priorities.

- **Develop an EPL scheduling technique**

Considering the scheduling techniques found in the literature, two EPL scheduling techniques were formulated and developed. Named the Tesla and Da Vinci solvers, these scheduling techniques have their roots in simulated annealing. The two methodologies and their respective solver parameters are summarised next.

Although the Tesla and Da Vinci solvers are simulated annealing solvers, they have dissimilar methodologies. The Tesla solver uses a flexible temperature schedule and slower cooling rate while the Da Vinci solver uses a faster cooling rate, fixed temperature schedule and a reheat strategy. With the flexible temperature schedule used in the Tesla solver, the temperature changes only after a consecutive number of non-improving moves are made. The aim was to allow the solver to effectively search all hills in the local search region before deciding which hills to explore further. On the other hand, at the end of the temperature schedule in the Da Vinci solver, a reheat strategy creates a new temperature schedule with reduced temperatures. The aim was to allow the solver to escape hills which it can get trapped in at low temperatures due to its faster cooling rate. In both cases, a hill-climbing add-on supplemented the solvers. Particularly with the Da Vinci solver, it was found that the hill-climber effectively produced better quality solutions. This is because it allowed the solver to escape small hills in the final hill area it was searching to find the best solution in this search vicinity.

Both solvers used the same neighbourhood search heuristic. A combination of three neighbourhood strategies found in the literature was used. The SwapRounds, SwapHomes and SwapTeams strategies were observed to be effective when used in tandem. In comparison, it was found that employing only a single strategy did not sufficiently explore the search space. Opportunities for improvement with regards to the search strategy are described later (Section 10.3).

To initiate the solvers, an initial trial solution was required. In this regard, since it is difficult to create manually, a polygon method described in the literature was used and is presented in Appendix F.

Table 10 and Table 11 (Section 6.2.5) show the solver parameters used in each of the Da Vinci and Tesla solvers. These parameters were set using a combination of trial and error and practice found in the literature.

- **Validate the EPL scheduling technique using three instances**

It was found in the results (Table 12 in Section 8.2) that the Tesla solver produced superior results in terms of solution quality and reliability compared to the Da Vinci solver for the 2013/14 season. Although the Da Vinci solver could not match the solution quality of the Tesla solver, it was found that the solution quality was at least comparable with that produced by the league (total cost gap of +4%).

The Tesla solver was then selected and validated by comparing its results to official results over three seasons, from the 2011/12 season to the 2013/14 season (Table 13 in Section 8.3). Over these three seasons, assuming that the weights used accurately depicted constraint priorities, solutions with at least 40% improvement in solution quality were found. However, the solving speed is slightly slower at 40 minutes compared to Atos's 30 minutes. Even though it is slightly slower, the solution quality is of more importance.

10.3 Recommendations

Three recommendations for further research into EPL scheduling are presented. A fourth recommendation concerning other real-world football scheduling instances are shown.

- **Limitations**

Limitations of this research were presented and justified in Section 6.1.2. For example, only the paired teams in the EPL were considered, while the league actually pairs teams across multiple divisions. It was justified since it is possible for other leagues to be scheduled around the significantly higher valued EPL league. Such limitations however, could be considered further.

- **Holistic system**

A more holistic model can be formulated. Referee assignment and timeslot assignment with the use of Balanced Tournament Design discussed in the literature (Section 3.3.3) are two such cases. As another case, one could incorporate transport route considerations which aim to minimise the pressure on certain routes.

- **Neighbourhood strategy**

One key aspect identified and discussion in Section 9.3 which is likely to significantly improve the solution quality of the results is the neighbourhood strategy. Anagnostopoulos et al. [44] and Riberiro and Urrutia [46] agree that the SwapRounds, SwapHomes and SwapTeams strategies cannot sufficiently explore the search space. Therefore, it is suggested that a strategy is formulated which uses ejection chains. Examples of such moves include the PartialSwapRounds and PartialSwapTeams strategies proposed by Anagnostopoulos et al. [44] and GameSwap by Riberiro and Urrutia [46].

- **Other real-world football scheduling instances**

It was highlighted in Section 3.4 that the nature of the EPL scheduling problem is different to other published real-world football sports scheduling problems. Further research may be conducted using the solving methodologies formulated in this research to solve other real-world instances. The results of those instances can then be compared to published results in the literature if it exists.

10.4 Conclusion

To conclude, this research presents a scheduling technique which answers the request by Nurmi et al. [3] to provide a framework and benchmark for one of the unsolved real-world sports scheduling problems. A simulated annealing technique was used and verified. Results were then validated over the last three seasons. Lastly, the EPL scheduling problem and solutions presented in this research (with its associated violations and travel distances) provide a maiden instance and benchmark for the real-world EPL scheduling problem.

11 REFERENCES

[1] Deloitte, “Football Finance - 2013 Annual Review - Sport Finance,” London, 2013.

[2] Premier League, “Fixture fact sheet,” London, 2012.

[3] K. Nurmi, D. Goossens, T. Bartsch, F. Bonomo, D. Briskorn, G. Duran, J. Kyngäs, J. Marenco, C. C. Ribeiro, F. Spieksma, S. Urrutia, R. Wolf-Yadlin, S.-I. Ao, H. Katagir, L. Xu, and A. H.-S. Chan, “A framework for a highly constrained sports scheduling problem,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists - IMECS 2010*, 2010, vol. III, pp. 14–28.

[4] G. Kendall, “Scheduling English football fixtures over holiday periods,” *Journal of the Operational Research Society*, vol. 59, no. 6, pp. 743–755, Mar. 2007.

[5] TheFA, “The FA Cup Round Dates,” 2013. [Online]. Available: http://www.thefa.com/TheFACup/More/fa-cup-round-dates. [Accessed: 15-Jun-2013].

[6] The Football League, “About the Competition.” [Online]. Available: http://www.capitalonecup.co.uk/competition/about-the-competition/. [Accessed: 16-Jun-2013].

[7] TheFA, “Club Admin,” 2013. [Online]. Available: http://www.thefa.com/thefacup/more/clubadmin. [Accessed: 15-Jun-2013].

[8] TheFA, “Cup Final Day Confirmed,” 2013. [Online]. Available: http://www.thefa.com/TheFACup/News/2013/Jun/fa-cup-final-date-2014.aspx. [Accessed: 19-Jun-2013].

[9] UEFA, “UEFA Europa League - Draws,” 2013. [Online]. Available: http://www.uefa.com/uefaeuropaleague/season=2014/draws/round=2000465/index.html. [Accessed: 16-Jun-2013].

[10] Premier League, “Frequently Asked Questions,” 2013. [Online]. Available: http://www.premierleague.com/content/premierleague/en-gb/fans/faqs.html. [Accessed: 11-Apr-2013].

[11] UEFA, “UEFA Super Cup - competition format,” 2013. [Online]. Available: http://www.uefa.com/uefasupercup/season=2013/competitionformat/index.html. [Accessed: 12-Jun-2013].

[12] FIFA, “Match Schedule,” 2013. [Online]. Available: http://www.fifa.com/mm/document/tournament/competition/02/09/75/34/fcwc2013-matchschedule-updated-03062013_neutral.pdf. [Accessed: 17-Jun-2013].

[13] J. Pearce, “Premier League rights sold to BT and BSkyB for £3bn,” *BBC*, 2012. [Online]. Available: http://www.bbc.co.uk/news/business-18430036. [Accessed: 18-Jun-2013].

[14] TheFA, “The FA Cup - Prize Fund,” 2012. [Online]. Available: http://www.thefa.com/TheFACup/More/prize-fund. [Accessed: 15-Jun-2013].

[15] TheFA and Deloitte, “The Financial Impact of the FA Cup,” 2012. [Online]. Available: http://www.thefa.com/TheFACup/News/2012/Jan/~/media/421387323A984557A3686BA18D383E6D.ashx. [Accessed: 20-Jun-2013].

[16] D. Conn, “Bradford City believe League Cup achievement is one for the underdogs,” *The Guardian*, 2013. [Online]. Available: http://www.guardian.co.uk/football/blog/2013/jan/23/bradford-city-league-cup-final. [Accessed: 20-Jun-2013].

[17] UEFA, “UEFA Champions League revenue distribution,” 2012. [Online]. Available: http://www.uefa.org/management/finance/news/newsid=1845591.html. [Accessed: 12-Jun-2013].

[18] UEFA, “Clubs benefit from Champions League revenue,” 2012. [Online]. Available: http://www.uefa.org/management/finance/news/newsid=1840934.html#clubs+benefit+from+revenue. [Accessed: 13-Jun-2013].

[19] UEFA, “UEFA Europa League revenue distribution - UEFA,” 2012. [Online]. Available: http://www.uefa.org/management/finance/news/newsid=1845777.html. [Accessed: 12-Jun-2013].

[20] UEFA, “UEFA Europa League club payments,” 2012. [Online]. Available: http://www.uefa.org/management/finance/news/newsid=1840941.html. [Accessed: 12-Jun-2013].

[21] Daily Mail, “Club World Cup: It’s big in Japan, and so is the prize money,” *Mail Online*, 2008. [Online]. Available: http://www.dailymail.co.uk/sport/football/article-1097020/Club-World-Cup-Its-big-Japan-prize-money.html. [Accessed: 19-May-2013].

[22] FIFA, “The FIFA Club World Cup in numbers,” 2012. [Online]. Available: http://www.fifa.com/tournaments/archive/clubworldcup/japan2012/news/newsid=1961123/. [Accessed: 17-Jun-2013].

[23] Premier League, “The perfect match: How the fixture list is compiled,” 2013. [Online]. Available: http://www.premierleague.com/en-gb/news/news/2013-14/jun/how-the-fixture-list-is-compiled.html. [Accessed: 25-Jun-2013].

[24] D. Worth, “Atos reveals the secrets to unlocking the football fixture conundrum,” *V3*, 2012. [Online]. Available: http://www.v3.co.uk/v3-uk/analysis/2184337/atos-reveals-secrets-unlocking-football-fixture-conundrum. [Accessed: 12-Nov-2013].

[25] EPFL, “How the fixtures are produced.” [Online]. Available: http://www.epfl-europeanleagues.com/how_the_fixtures_are_produced.htm. [Accessed: 19-Jun-2013].

[26] Football Supporters’ Federation, “All Tomorrow’s (Fixture Working) Parties,” 2012.

[27] J. Riach, “How the Premier League and Football League fixtures are compiled,” *The Guardian*, 2013. [Online]. Available: http://www.theguardian.com/football/2013/jun/15/fixtures-premier-league-football-league-compiled. [Accessed: 18-Jun-2013].

[28] G. Kendall, S. Knust, C. C. Ribeiro, and S. Urrutia, “Scheduling in sports: An annotated bibliography,” *Computers & Operations Research*, vol. 37, no. 1, pp. 1–19, Jan. 2010.

[29] S. Knust, “Classification of Literature on Sports Scheduling (maintained by Sigrid Knust).” [Online]. Available: http://www.inf.uos.de/knust/sportssched/sportlit_class/sportlit_class.html. [Accessed: 24-Jun-2013].

[30] R. V. Rasmussen and M. A. Trick, “Round robin scheduling – a survey,” *European Journal of Operational Research*, vol. 188, no. 3, pp. 617–636, Aug. 2008.

[31] T. Bartsch, A. Drexl, and S. Kröger, “Scheduling the professional soccer leagues of Austria and Germany,” *Computers & Operations Research*, vol. 33, no. 7, pp. 1907–1937, Jul. 2006.

[32] D. de Werra, “Some models of graphs for scheduling sports competitions,” *Discrete Applied Mathematics*, vol. 21, no. 1, pp. 47–65, Sep. 1988.

[33] J. Dinitz, “Designing Schedules for Leagues and Tournaments,” in *Graph Theory Day 48*, 2004.

[34] K. Easton, G. Nemhauser, and M. Trick, “The Traveling Tournament Problem Description and Benchmarks,” in *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming - CP 2001*, 2001, pp. 580–584.

[35] D. R. Goossens and F. C. R. Spieksma, “The Carryover Effect Does Not Influence Football Results,” *Journal of Sports Economics*, vol. 13, no. 3, pp. 288–305, Apr. 2011.

[36] D. Briskorn, *Sports Leagues Scheduling*, vol. 603. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[37] D. Briskorn, A. Drexl, and F. Spieksma, “Round robin tournaments and three index assignments,” *4OR*, vol. 8, no. 4, pp. 365–374, Mar. 2010.

[38] T. Walsh and I. . Gent, “CSPLib: a problem library for constraints,” 1999. [Online]. Available: http://csplib.cs.strath.ac.uk/. [Accessed: 16-Aug-2013].

[39] J. Dinitz and M. Dinitz, “Enumeration of balanced tournament designs on 10 points,” *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 52, pp. 51–63, 2005.

[40] J. A. M. Schreuder, “Combinatorial aspects of construction of competition Dutch Professional Football Leagues,” *Discrete Applied Mathematics*, vol. 35, no. 3, pp. 301–312, Mar. 1992.

[41] F. Hillier and G. Lieberman, *Introduction to Operations Research*, 9th ed. Singapore: McGraw Hill, 2010, pp. 607–646.

[42] A. Schaerf, “Scheduling Sport Tournaments using Constraint Logic Programming,” *Constraints*, vol. 4, no. 1, pp. 43–65, 1999.

[43] J. Kyngäs, “Solving Challenging Real-World Scheduling Problems,” University of Turku, 2011.

[44] A. Anagnostopoulos, L. Michel, P. van Hentenryck, and Y. Vergados, “A simulated annealing approach to the traveling tournament problem,” *Journal of Scheduling*, vol. 9, no. 2, pp. 177–193, Apr. 2006.

[45] A. Lim, B. Rodrigues, and X. Zhang, “A simulated annealing and hill-climbing algorithm for the traveling tournament problem,” *European Journal of Operational Research*, vol. 174, no. 3, pp. 1459–1478, Nov. 2006.

[46] C. C. Ribeiro and S. Urrutia, “Heuristics for the mirrored traveling tournament problem,” *European Journal of Operational Research*, vol. 179, no. 3, pp. 775–787, Jun. 2007.

[47] F. Della Croce and D. Oliveri, “Scheduling the Italian Football League: an ILP-based approach,” *Computers & Operations Research*, vol. 33, no. 7, pp. 1963–1974, Jul. 2006.

[48] D. Goossens and F. Spieksma, “Scheduling the Belgian Soccer League,” *Interfaces*, vol. 39, no. 2, pp. 109–118, Apr. 2009.

[49] W. D. Wallis, “One-Factorizations of Graphs: Tournament Applications,” *The College Mathematics Journal*, vol. 18, no. 2, p. 116, Mar. 1987.

[50] Clay Mathematics Institute, “P vs NP Problem Clay Mathematics Institute.” [Online]. Available: http://www.claymath.org/millenium-problems/p-vs-np-problem. [Accessed: 12-Aug-2013].

[51] G. Kendall, “RE: Scheduling English Football Fixtures over Holiday Periods.” Personal email (Jun. 24, 2013).

[52] M. Park and Y.-D. Kim, “A systematic procedure for setting parameters in simulated annealing algorithms,” *Computers & Operations Research*, vol. 25, no. 3, pp. 207–217, Mar. 1998.

[53] R. J. Willis and B. J. Terrill, “Scheduling the Australian State Cricket Season Using Simulated Annealing,” *Journal of the Operational Research Society*, vol. 45, no. 3, pp. 276–280, Mar. 1994.

[54] M. Wright, "Scheduling fixtures for Basketball New Zealand," *Computers & Operations Research*, vol. 33, no. 7, pp. 1875–1893, Jul. 2006.

[55] Wikipedia, "2013–14 Premier League - Wikipedia, the free encyclopedia," 2013. [Online]. Available: http://en.wikipedia.org/wiki/2013%E2%80%9314_Premier_League. [Accessed: 03-May-2013].

[56] UK Government, "UK Bank Holidays." [Online]. Available: https://www.gov.uk/bank-holidays. [Accessed: 25-May-2013].

[57] UEFA, "UEFA Champions League access list." [Online]. Available: http://www.uefa.com/uefachampionsleague/season=2014/accesslist/index.html. [Accessed: 15-May-2013].

[58] UEFA, "UEFA Europa League access list." [Online]. Available: http://www.uefa.com/uefaeuropaleague/season=2014/accesslist/. [Accessed: 16-May-2013].

APPENDIX A
GEOGRAPHIC POSITIONS OF 2013/14 EPL TEAMS

The points on the maps below indicate the spread of teams across the United Kingdom. The close geographical locations of some teams can be seen in these maps and justifies why some teams are 'paired'.

Teams in the UK

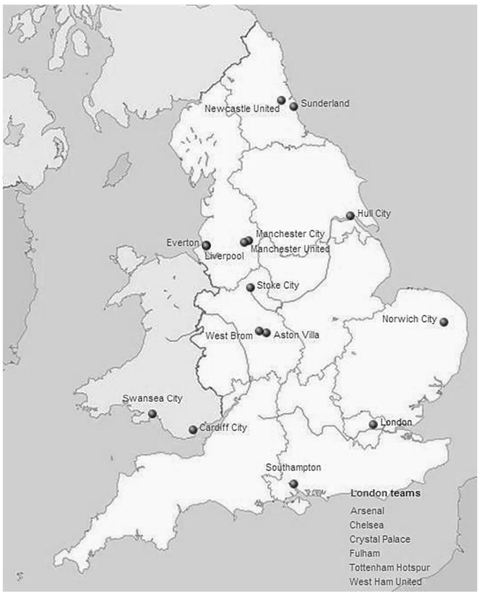

Figure A1: Location of EPL teams for the 2013/14 season [55]

Teams based in Greater London

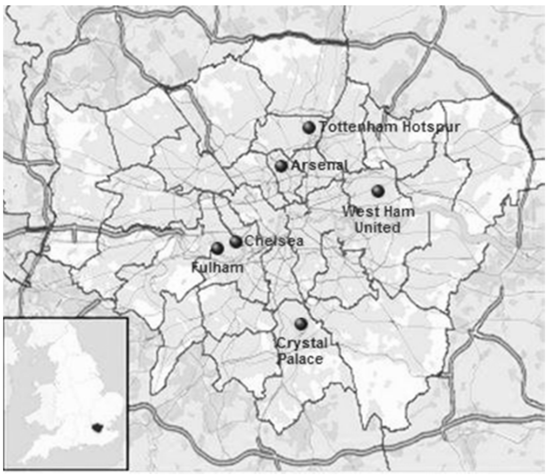

Figure A2: EPL teams based in Greater London for the 2013/14 season [55]

APPENDIX B

KEY DATES AFFECTING EPL SCHEDULING

In this appendix are the key dates and corresponding slots which must be considered in the scheduling of the EPL. Three seasons worth of data are presented in the tables.

Bank Holidays

Table B1: UK Bank Holidays [56]

Holiday	2013/2014	(slot)	2012/2013	(slot)	2011/2012	(slot)
Summer Bank holiday	26 Aug 13	(2)	27 Aug 12	(2)	29 Aug 11	(3)
Boxing Day	26 Dec 13	(18)	26 Dec 12	(19)	26 Dec 11	(18)
New Year's Day	01 Jan 14	(20)	01 Jan 13	(21)	02 Jan 12	(20)
Good Friday	18 Apr 14	(35)	29 Mar 13	(31)	06 Apr 12	(32)
Easter Monday	21 Apr 14	(35)	01 Apr 13	(31)	09 Apr 12	(32)
Early May Bank holiday	05 May 14	(37)	06 May 13	(36)	07 May 12	(37)

It is important to note that Good Friday and Easter Monday only influence one slot since these public holidays are over one weekend. Also noteworthy is that the slots for the holiday period fixtures (i.e. Boxing Day and New Year's Day) are not on the same slots every year.

Domestic Cup tournaments

The FAC competition commences with the preliminary rounds in mid-August with the EPL teams only entering in the third proper round of the competition.

Table B2: FAC round dates [5]

Round	2013/2014	(slot)	2012/2013	(slot)	2011/2012	(slot)
Third Proper	04 Jan 14	(20-21)	05-06 Jan 13	(21-22)	07 Jan 12	(20-21)
Fourth Proper	25 Jan 14	(22-23)	26-27 Jan 13	(23-24)	28 Jan 12	(22-23)
Fifth Proper	15 Feb 14	(26-27)	16-18 Feb 13	(26-27)	18 Feb 12	(25-26)
¼ finals	08 Mar 14	(29-30)	09-10 Mar 13	(29-30)	17 Mar 12	(29-30)
½ finals	12-13 Apr 14	(34-35)	13-14 Apr 13	(33-34)	14-15 Apr 12	(34-35)
Final	17 May 14	(N/A)	11 May 13	(N/A)	05 May 12	(N/A)

The final is contested by only two teams and for this reason is not considered in the constraint formulation (C105). If it were considered, since the C105 constraint assumes that all teams' progress, penalties would be unnecessarily incurred.

With regards to FAC games which result in draws and consequently replays, the Premier League takes responsibility for postponing affected EPL fixtures (usually to midweek).

European tournaments

Different criteria exist for how, and at what stage, teams enter the two European competitions. The two tables below show the dates and corresponding slots for the UCL and UEL respectively. Although similar, the affected slots for the UCL and UEL can be different.

Table B3: UCL round dates [57]

Round	2013/2014	(slot)	2012/2013	(slot)	2011/2012	(slot)
Group, match 1	17-18 Sep 13	(4-5)	18-19 Sep 12	(4-5)	13-14 Sep 11	(4-5)
Group, match 2	01-02 Oct 13	(6-7)	02-03 Oct 12	(6-7)	27-28 Sep 11	(6-7)
Group, match 3	22-23 Oct 13	(8-9)	23-24 Oct 12	(8-9)	18-19 Oct 11	(8-9)
Group, match 4	05-06 Nov 13	(10-11)	06-07 Nov 12	(10-11)	01-02 Nov 11	(10-11)
Group, match 5	26-27 Nov 13	(12-13)	20-21 Nov 12	(12-13)	22-23 Nov 11	(12-13)
Group, match 6	10-11 Dec 13	(15-16)	04-05 Dec 12	(15-16)	06-07 Dec 11	(14-15)
Rnd of 16, leg 1	18-19/25-26 Feb 14	(26-27/27-28)	12-13/19-20 Feb 13	(26-27)	14-15/21-22 Feb 12	(25-26)
Rnd of 16, leg 2	11-12/18-19 Mar 14	(29-30/30-31)	05-06/12-13 Mar 13	(28-29/29-30)	06-07/13-14 Mar 12	(27-28/28-29)
¼ finals, leg 1	01-02 Apr 14	(32-33)	02-03 Apr 13	(31-32)	27-28 Mar 12	(30-31)
¼ finals, leg 2	08-09 Apr 14	(33-34)	09-10 Apr 13	(32-33)	03-04 Apr 12	(31-32)
½ finals, leg 1	22-23 Apr 14	(35-36)	23-24 Apr 13	(34-35)	17-18 Apr 12	(34-35)
½ finals, leg 2	29-30 Apr 14	(36-37)	30 Apr-1 May 13	(35-36)	24-25 Apr 12	(35-36)
Final	24 May 14	(N/A)	25 May 13	(N/A)	19 May 12	(N/A)

Table B4: UEL round dates [58]

Round	2013/2014	(slot)	2012/2013	(slot)	2011/2012	(slot)
Group, match 1	19 Sep 13	(4-5)	20 Sep 12	(4-5)	15 Sep 11	(4-5)
Group, match 2	03 Oct 13	(6-7)	04 Oct 12	(6-7)	29 Sep 11	(6-7)
Group, match 3	24 Oct 13	(8-9)	25 Oct 12	(8-9)	20 Oct 11	(8-9)
Group, match 4	07 Nov 13	(10-11)	08 Nov 12	(10-11)	03 Nov 11	(10-11)
Group, match 5	28 Nov 13	(12-13)	22 Nov 12	(12-13)	30-1 Dec 11	(13-14)
Group, match 6	12 Dec 13	(15-16)	06-07 Dec 12	(15-16)	14-15 Dec 11	(15-16)
Rnd of 32, leg 1	20 Feb 14	(26-27)	14 Feb 13	(26-27)	14-16 Feb 12	(25-26)
Rnd of 32, leg 2	27 Feb 14	(27-28)	21 Feb 13	(26-27)	22-23 Feb 12	(25-26)
Rnd of 16, leg 1	13 Mar 14	(29-30)	07 Mar 13	(28-29)	08 Mar 12	(27-28)
Rnd of 16, leg 1	20 Mar 14	(30-31)	14 Mar 13	(29-30)	15 Mar 12	(28-29)
¼ finals, leg 1	03 Apr 14	(32-33)	04 Apr 13	(31-32)	29 Mar 12	(30-31)
¼ finals, leg 2	10 Apr 14	(33-34)	11 Apr 13	(32-33)	05 Apr 12	(31-32)
½ finals, leg 1	24 Apr 14	(35-36)	25 Apr 13	(34-35)	19 Apr 12	(34-35)
½ finals, leg 2	01 May 14	(36-37)	02 May 13	(35-36)	26 Apr 12	(35-36)
Final	14 May 14	(N/A)	15 May 13	(N/A)	09 May 12	(N/A)

It was stated in Section 6.1.2 that, like the FAC, it is assumed that the EPL teams will progress to the latter stages of the European competitions. Therefore, constraints C102c and C102e as well as C102d and C102f hold until the semi-final rounds. Justifying the decision to leave the finals out of the calculation is that the UCL finals takes place after the league concludes while the UEL usually takes place before the last game of the season and has little effect on their European prospects.

Club away requests

The table which follows is used for constraint C04 which concerns the away requests of clubs (or police request). It must be stressed that the events identified here are not a definitive list but rather a list of those events which have been identified from the literature.

Table B5: Club Away Request

Event (affected clubs)	2013/2014	(slot)	2012/2013	(slot)	2011/2012	(slot)
Manchester Pride (Manchester clubs)	23-26 Aug 13	(2)	24-27 Aug 12	(2)	27-29 Aug 11	(3)
Notting Hill Carnival (Chelsea, Fulham, QPR)	25 Aug 13	(2)	26 Aug 12	(2)	28 Aug 11	(3)
Cardiff Mardi Gras (Cardiff)	31 Aug 13	(3)	-		-	
Bupa Great North Run (Newcastle and Sunderland)	15 Sep 13	(4)	16 Sep 12	(4)	18 Sep 11	(5)
Sunderland Pride (Sunderland)	01 Sep 13	(3)	23 Sep 12	(5)	25 Sep 11	(6)
Labour Party Conference 2012 (Manchester clubs) 2011 (Liverpool, Everton)	-		30 Sep - 04 Oct 12	(6)	25-29 Sep 11	(6)
Conservative Party 2013 (Manchester clubs) 2012 (Aston Villa, WBA) 2011 (Manchester clubs)	29 Sep 13	(6)	07-10 Oct 12	(7)	02-05 Oct 11	(7)
Boat race (Chelsea, Fulham)	06 Apr 14	(33)	31 Mar 13	(31)	07 Apr 12	(32)
Crabbie's Grand National (Liverpool, Everton)	03-05 Apr 14	(33)	04-06 Apr 13	(32)	11-14Apr 12	(34)

Published domestic dates

The fixture list for the 2013/14 EPL was confirmed and announced on 19 June 2013. Meanwhile, the FAC dates had already been presented in the FA Competitions Department bulletin on 10 May 2013.

Published European cups dates

Dates for 2013/14 season of the two European cups, the UCL and UEL, were uploaded on the UEFA site on 14 May 2013.

APPENDIX C

TEAM CODE AND STADIUMS

The table below lists the stadiums of the 2013/14, 2012/13 and 2011/12 EPL teams. These stadia names where entered into Google maps to determine the relative driving distances. To avoid cluster, only unique stadia names are shown (i.e. the names of stadia are provided if it is not shown before in the table).

Table C1: Team code and stadia

Team code	2013/14 team	Stadium	2012/13 team	Stadium	2011/12 team	Stadium
t_1	Man Utd.	Old Trafford	Man City		Man Utd.	
t_2	Man City	Etihad	Man Utd.		Chelsea	
t_3	Chelsea	Stamford Bridge	Arsenal		Man City	
t_4	Arsenal	Emirates	Tottenham		Arsenal	
t_5	Tottenham	White Hart Lane	Newcastle		Tottenham	
t_6	Everton	Goodison Park	Chelsea		Liverpool	
t_7	Liverpool	Anfield	Everton		Everton	
t_8	West Brom.	The Hawthorns	Liverpool		Fulham	
t_9	Swansea	Liberty Stadium	Fulham		Aston Villa	
t_{10}	West Ham	Boleyn Ground	West Brom.		Sunderland	
t_{11}	Norwich City	Carrow Road	Swansea		West Brom.	
t_{12}	Fulham	Craven Cottage	Norwich City		Newcastle	
t_{13}	Stoke City	Britannia	Sunderland		Stoke City	
t_{14}	Southampton	St Mary's	Stoke City		Bolton	Reebok
t_{15}	Aston Villa	Villa Park	Wigan Athletic	DW Stadium	Blackburn	Ewood Park
t_{16}	Newcastle	St James' Park	Aston Villa		Wigan	
t_{17}	Sunderland	Stadium of Light	QPR	Loftus Road	Wolverhampton	Molineux
t_{18}	Hull City	KC Stadium	West Ham		Norwich City	
t_{19}	Cardiff City	Cardiff City	Reading	Madejski	QPR	
t_{20}	Crystal Palace	Selhurst Park	Southampton		Swansea	

APPENDIX D

PRE-SOLVER BIP (HOLIDAY PERIOD INTEGER PROGRAM)

A binary integer program was formulated to find the optimal distance **(minimised)** which teams would travel over the two holiday period fixtures (i.e. Boxing Day and New Year's Day fixtures). This is necessary since the simulated annealing program compares the distance travelled in a given candidate solution to that of the optimal distance.

Matlab's built in binary integer program (bintprog) can be used to solve this problem but can take hours to solve. Alternatively, one could use the CPLEX plug-in which solves the problem in under a second. Matlab requires input of the form $Ax = b$ and/or $Ax \leq b$ where A represents the binary constraint matrix, x represents the binary output vector and b the constant vector values of the constraints.

Objective

The linear objective of the problem is to minimise the distance travelled.

$$\delta = Z_{min} = D_{i,j}.X_{i,j} + D_{i,j}.Y_{i,j} \text{ (array-wise matrix multiplication)}$$

$$Z_{min} = d_{1,1}x_{1,1} + d_{1,2}x_{1,2} + \ldots + d_{20,20}x_{20,20} + d_{1,1}y_{1,1} + d_{1,2}y_{1,2} + \ldots + d_{20,20}y_{20,20} \text{ (vector form)}$$

Parameters

The distances between each team (measured in kilometres) were found by determining the relative driving distance between each stadium using Google Maps (Appendix C).

The parameter used to represent these distances is shown in Table D1 with the respective values for the 2013/14 season shown in Table D2.

Table D1: Driving distances parameter matrix (D)

i, j	t_1	t_2	t_3-t_{18}	t_{19}	t_{20}
t_1	$d_{1,1}$	$d_{1,2}$	…	$d_{1,19}$	$d_{1,20}$
t_2	$d_{2,1}$	$d_{2,2}$	…	$d_{2,19}$	$d_{2,20}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$d_{19,1}$	$d_{19,2}$	…	$d_{19,19}$	$d_{19,20}$
t_{20}	$d_{20,1}$	$d_{20,2}$	…	$d_{20,19}$	$d_{20,20}$

$$d_{i,j} \geq 0$$

Table D2: Driving distances between stadiums [km]

i, j	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
t_1	M	M	M	M	M	M	53	135	379	341	392	338	81	368	141	243	240	167	319	352
t_2	M	M	M	M	M	M	58	133	377	339	314	338	80	367	141	229	226	153	319	352
t_3	M	M	M	M	M	M	354	210	300	20	195	M	260	125	202	451	448	347	241	14
t_4	M	M	M	M	M	M	336	195	312	15	180	18	250	136	184	450	447	337	251	21
t_5	M	M	M	M	M	M	340	200	318	19	174	23	253	152	187	451	447	303	258	26
t_6	M	M	M	M	M	M	M	154	398	360	411	358	100	385	157	282	279	203	335	368
t_7	53	58	354	336	340	M	M	150	394	356	407	353	97	383	156	278	275	203	335	367
t_8	135	133	210	195	200	154	150	M	246	214	266	83	64	238	M	341	338	230	186	223
t_9	379	377	300	312	318	398	394	246	M	319	500	300	307	284	251	579	575	469	M	313
t_{10}	341	339	20	15	19	360	356	214	319	M	174	23	269	145	204	450	447	303	259	25
t_{11}	392	314	195	180	174	411	407	266	500	174	M	204	322	327	255	414	412	245	441	200
t_{12}	338	338	M	18	23	358	353	83	300	23	204	M	267	124	201	458	455	346	240	16
t_{13}	81	80	260	250	253	100	97	64	307	269	322	267	M	297	70	315	238	204	248	281
t_{14}	368	367	125	136	152	385	383	238	284	145	327	124	297	M	229	521	517	408	224	147
t_{15}	141	141	202	184	187	157	156	M	251	204	255	201	70	229	M	330	327	217	191	215
t_{16}	243	229	451	450	451	282	278	341	579	450	414	458	315	521	330	M	M	229	520	473
t_{17}	240	226	448	447	447	279	275	338	575	447	412	455	238	517	327	M	M	225	517	470
t_{18}	167	153	347	337	303	203	203	230	469	303	245	346	204	408	217	229	225	M	407	360
t_{19}	319	319	241	251	258	335	335	186	M	259	441	240	248	224	191	520	517	407	M	254
t_{20}	352	352	14	21	26	368	367	223	313	25	200	16	281	147	215	473	470	360	254	M

It is important to note that the diagonal has been assigned a Big-M distance of 1×10^4 km since a team cannot compete against itself.

Variables

Below are the two holiday period matrices. For visualisation purposes, the New Year's Day matrix is represented by a Y matrix when in fact is a continuation of the X matrix (i.e. $y_{1,1} = x_{21,1}$ etc.).

Table D3: Boxing Day variable matrix (X)

i, j	t_1	t_2	t_3-t_{18}	t_{19}	t_{20}
t_1	$x_{1,1}$	$x_{1,2}$	…	$x_{1,19}$	$x_{1,20}$
t_2	$x_{2,1}$	$x_{2,2}$	…	$x_{2,19}$	$x_{2,20}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$x_{19,1}$	$x_{19,2}$	…	$x_{19,19}$	$x_{19,20}$
t_{20}	$x_{20,1}$	$x_{20,2}$	…	$x_{20,19}$	$x_{20,20}$

Table D4: New Year's Day variable matrix (Y)

i, j	t_1	t_2	t_3-t_{18}	t_{19}	t_{20}
t_1	$y_{1,1}$	$y_{1,2}$	…	$y_{1,19}$	$y_{1,20}$
t_2	$y_{2,1}$	$y_{2,2}$	…	$y_{2,19}$	$y_{2,20}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$y_{19,1}$	$y_{19,2}$	…	$y_{19,19}$	$y_{19,20}$
t_{20}	$y_{20,1}$	$y_{20,2}$	…	$y_{20,19}$	$y_{20,20}$

As per the nature of binary integer programming, all variables are binary:

$$x_{i,j}, y_{i,j} = \begin{cases} 1, & if\ team\ t_i\ plays\ at\ home\ in\ slot\ s_j \\ 0, & otherwise \end{cases}$$

Constraints

Linear equality and linear inequality constraints of the form $Ax = b$ and $Ax \leq b$ respectively are shown. The constraints were formulated to ensure that a feasible solution would be calculated.

<u>A) Linear inequality constraints of the form *A* ≤ *b* are listed:</u>

C12c. Teams cannot have consecutive home (or away) holiday period fixtures.

$$\sum_{j=1} x_{i,j} + y_{i,j} \leq 1 \qquad \forall\, i$$

Conversely, a team may only play away on one of the two slots.

$$\sum_{i=1} x_{i,j} + y_{i,j} \leq 1 \qquad \forall\, j$$

C108. The holiday period fixtures on Boxing Day should not be repeated on New Year's Day (i.e. if $x_{5,1} = 1$ then $y_{5,1} \neq 1$).

$$x_{i,j} + y_{j,i} \leq 1 \qquad \forall\, i, j$$

<u>B) Next, linear equality constraints of the form *A* = *b* are listed:</u>

Each team plays exactly one fixture per slot.

$$\sum_{i=1} x_{i,j} + x_{j,i} = 1 \qquad \forall\, j$$

$$\sum_{i=1} y_{i,j} + y_{j,i} = 1 \qquad \forall\, j$$

For $n = 20$ teams, there will be exactly $n/2 = 10$ fixtures per holiday period slot.

$$\sum x_{i,j} = 10 \qquad \forall\, i, j$$

$$\sum y_{i,j} = 10 \qquad \forall\, i, j$$

C) Big-M

Other linear constraints include constraints C102b and C103. These constraints can be mathematically shown as $A = 0$. However, in this case, Big-M is used instead to simplify coding.

C102b. Avoid strength fixtures within the holiday period.

> Here, since the strength teams are defined as teams' t_1 to t_6 (see Table 7 in Section 5.4), the top left square of the driving distance table, Table D2, is assigned a Big-M distance.

C103. Paired teams should not be assigned to holiday period fixtures.

> Similarly, paired teams (Table 7) are assigned a Big-M distance in the Table D2.

APPENDIX E
EPL SCHEDULING PROBLEM (MATHEMATICAL FORMULATION)

A simulated annealing approach was used to guide the search for a good solution. The total cost assigned to a solution was found by combining the penalties associated with the home-away pattern set (*HAP Set*) with those associated to the timetable set (*T Set*).

Objective

The objective is to **minimise the total cost** incurred by a particular candidate solution (with no hard constraint violations which would result in an undesirable or infeasible solution).

$$Z_{min} = \sum \text{penalties (Such that the cumulated hard constraint penalties = 0)}$$

Parameters

It is important to note that the parameters defined here are for the 2013/14 EPL season. Parameter data can be found in Section 5.4 and in Appendix B.

ω	denotes the penalties[7] = ω_{Cx} (i.e. penalty for violating C113 is ω_{C13}).
δ	denotes the optimal holiday period driving distance[8].
γ_{px}, γ_{py}, $\lvert\gamma_p\rvert$	denotes the paired team set one {t_1, $\mathbf{t_3}$, t_4, t_6, t_8, t_9, t_{16}}, the paired team set two {t_2, $\mathbf{t_{12}}$, t_5, t_7, t_{15}, t_{19}, t_{17}} and the number of paired teams respectively (i.e. in this case, γ_{p2x} and γ_{p2y} are paired teams and therefore team t_3 and team t_{12} are paired teams in a set of seven paired teams).
ε, $\lvert\varepsilon\rvert$	denotes the strength team set {t_1, t_2, t_3, t_4, t_5, t_6} and the number of strength teams.
η_t, η_f, $\lvert\eta_t\rvert$	denotes the away request team set {$\mathbf{t_1}$, $\mathbf{t_1}$, t_2, t_2, t_3, t_3, t_6, t_7, t_{12}, t_{12}, t_{16}, t_{17}, t_{17}, t_{19}}, the away request slot set {$\mathbf{s_2}$, $\mathbf{s_6}$, s_2, s_6, s_2, s_{33}, s_{33}, s_{33}, s_2, s_{33}, s_4, s_3, s_4, s_3} and the number of requests (i.e. team t_1 requests an away fixture for slots s_2 and s_6).
φ_t, $\lvert\varphi_t\rvert$	denotes the UCL team set {t_1, t_2, t_3, t_4} and the number of UCL teams.

[7] A cost is assigned for each constraint violation. Definitions of the constraints, their codes and respective weights can be seen in Section 5.3.
[8] Calculated in the Pre-Solver BIP (Appendix D).

φ_{fx}, φ_{fy}, $|\varphi_f|$ denotes the UCL fixture slot one set {s_4, s_6, s_8, s_{10}, s_{12}, s_{15}, s_{26}, s_{27}, s_{29}, s_{30}, s_{32}, s_{33}, s_{35}, s_{36}}, the UCL fixture slot two set {s_5, s_7, s_9, s_{11}, s_{13}, s_{16}, s_{27}, s_{28}, s_{30}, s_{31}, s_{33}, s_{34}, s_{36}, s_{37}} and the number of UCL fixtures.

λ_t, $|\lambda_t|$ denotes the UEL team set {t_5, t_9} and the number of UEL teams.

λ_{fx}, λ_{fy}, $|\lambda_{fx}|$ denotes the UEL fixture slot one set {s_4, s_6, s_8, s_{10}, s_{12}, s_{15}, s_{26}, s_{27}, s_{29}, s_{30}, s_{32}, s_{33}, s_{35}, s_{36}}, the UEL fixture slot two set {s_5, s_7, s_9, s_{11}, s_{13}, s_{16}, s_{27}, s_{28}, s_{30}, s_{31}, s_{33}, s_{34}, s_{36}, s_{37}} and the size of UEL fixtures.

μ_{fx}, μ_{fy}, $|\mu_{fx}|$ denotes the FAC slot one set {s_{20}, s_{22}, s_{26}, s_{29}, s_{34}}, the FAC slot two set {s_{21}, s_{23}, s_{27}, s_{30}, s_{35}} and the size of FAC slots.

κ, $|\kappa|$ denotes the Bank Holiday slot set {s_2, s_{18}, s_{20}, s_{35}, s_{37}} and the number of bank holiday slots.

σ_1, σ_2 denotes the equations $\sigma_1 = truncate\ [|\kappa|/2]$ and $\sigma_2 = integer\ [|\kappa|/2]$ (i.e. if $|\kappa| = 4$, then $\sigma_1 = 2$ and $\sigma_2 = 2$ but if $|\kappa| = 5$, then $\sigma_1 = 2$ and $\sigma_2 = 3$).

τ, $|\tau|$ denotes the holiday period slots {s_{18}, s_{20}} and the holiday period size.

Ω, $|\Omega|$ denotes the reverse slots {s_{14}, s_{25}} and the reverse slot size.

Variables

Table E1: Schedule variable matrix (F)

t, s	s_1	s_2	s_3-s_{36}	s_{37}	s_{38}
t_1	$f_{1,1}$	$f_{1,2}$	…	$f_{1,37}$	$f_{1,38}$
t_2	$f_{2,1}$	$f_{2,2}$	…	$f_{2,37}$	$f_{2,38}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$f_{19,1}$	$f_{19,2}$	…	$f_{19,37}$	$f_{19,38}$
t_{20}	$f_{20,1}$	$f_{20,2}$	…	$f_{20,37}$	$f_{20,38}$

This matrix is of size [20x38] since there are $n = 20$ teams in the league with a total of $2n\text{-}2 = 38$ games per team (19 home and 19 away fixtures in a 2RR format). Integer variable $f_{t,s}$ represents the opposition for team t in a particular slot s with a positive and negative integer representing a home, or away, fixture respectively[9]:

$$f_{t,s} = integer$$

[9] Example: $f_{19,2} = +5$ indicates that team 19 has a home fixture (away fixture if -ve) against team t_5 in slot s_2

Constraints

For the purpose of analysis, the variables from Table E1 were translated into binary integer *bf* (Table E2) and absolute *af* (Table E3) versions. Respectively, the *HAP Set* and *T Set* constraints use these translations. The *HAP Set* constraints, *T Set* constraints and constraints regarding distance travelled in the holiday period slots are shown below.

A) *HAP Set* constraints are listed:

Table E2: Schedule HAP Set variable matrix (binary~F)

t, s	s_1	s_2	s_3-s_{36}	s_{37}	s_{38}
t_1	$bf_{1,1}$	$bf_{1,2}$	…	$bf_{1,37}$	$bf_{1,38}$
t_2	$bf_{2,1}$	$bf_{2,2}$	…	$bf_{2,37}$	$bf_{2,38}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$bf_{19,1}$	$bf_{19,2}$	…	$bf_{19,37}$	$bf_{19,38}$
t_{20}	$bf_{20,1}$	$bf_{20,2}$	…	$bf_{20,37}$	$bf_{20,38}$

A home fixture is represented by the number one while an away fixture is represented by a zero[10].

$$bf_{t,s} = \begin{cases} 1, & if\ team\ t\ plays\ at\ home\ in\ slot\ s \\ 0, & otherwise \end{cases}$$

C13. A team cannot have more than two consecutive home games.

$$if \sum_{s=1}^{36} bf_{t,s} + bf_{t,s+1} + bf_{t,s+2} = 3\,, then\ count\ violations \cdot \omega_{C13} \qquad \forall\, t$$

C14. A team cannot have more than two consecutive away games.

$$if \sum_{s=1}^{36} bf_{t,s} + bf_{t,s+1} + bf_{t,s+2} = 0\,, then\ count\ violations \cdot \omega_{C14} \qquad \forall\, t$$

C12a. Avoid a break on the first two slots of the season.

$$if \sum bf_{t,1} + bf_{t,2} \neq 1\,, then\ count\ violations \cdot \omega_{C12a} \qquad \forall\, t$$

C12b. Avoid a break on the last two slots of the season.

$$if \sum bf_{t,37} + bf_{t,38} \neq 1\,, then\ count\ violations \cdot \omega_{C12b} \qquad \forall\, t$$

[10] Example: if $f_{19,2} = +5$, then $bf_{19,2} = 1$ since team t_{19} has a home fixture in slot s_2, but if $f_{19,2} = -5$, then $bf_{19,2} = 0$ since this would indicate that team t_{19} has an away fixture in slot s_2.

C12c. Teams cannot have consecutive home (or away) holiday period fixtures.

$$if \sum_{\tau=\tau_1}^{|\tau|} bf_{t,\tau} \neq 1\,, then\ count\ violations \cdot \omega_{C101} \qquad \forall\ t$$

C106. Each team should have an equal number of home and away Bank Holiday fixtures, assuming an even number are scheduled.

$$if \sum_{\kappa=\kappa_1}^{|\kappa|} bf_{t,\kappa} \neq \sigma_1 \text{ or } \sigma_2\,, then\ count\ violations \cdot \omega_{C106} \qquad \forall\ t$$

C105. EPL matches either side of a FAC fixture should not both be away from home.

$$if \sum_{\mu_f=\mu_{f1}}^{|\mu_f|} bf_{t,\mu_{fx}} + bf_{t,\mu_{fy}} \neq 1\,, then\ count\ violations \cdot \omega_{C105} \qquad \forall\ t$$

C04. Some teams should not necessarily play at home in certain slots.

$$if \sum_{\eta_f=\eta_{f1}}^{|\eta_f|} bf_{\eta_t,\eta_f} \neq 0, then\ count\ violations \cdot \omega_{C04} \qquad \forall\ \eta_t$$

C07. There should be at least zero and at most one home game between paired teams in the same slot (i.e. complementary pattern vectors in the *HAP Set* for paired teams).

$$if \sum_{s=1}^{s=38} bf_{\gamma_{px},s} + bf_{\gamma_{py},s} = 2, then\ count\ violations \cdot \omega_{C07} \qquad \forall\ \gamma_p$$

C110. The sum of the binary *HAP Set* for each team must equal $|s|/2$.

$$if \sum_{s=1}^{38} bf_{t,s} \neq 19, then\ count\ violations \cdot \omega_{C110} \qquad \forall\ t$$

C111. The sum of the binary *HAP Set* for each slot must equal $n/2$.

$$if \sum_{t=1}^{20} bf_{t,s} \neq 10, then\ count\ violations \cdot \omega_{C111} \qquad \forall\ s$$

B) *T Set* constraints are listed:

Table E3: Schedule T Set variable matrix (absolute~F)

t, s	s_1	s_2	s_3-s_{36}	s_{37}	s_{38}
t_1	$af_{1,1}$	$af_{1,2}$	…	$af_{1,37}$	$af_{1,38}$
t_2	$af_{2,1}$	$af_{2,2}$	…	$af_{2,37}$	$af_{2,38}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$af_{19,1}$	$af_{19,2}$	…	$af_{19,37}$	$af_{19,38}$
t_{20}	$af_{20,1}$	$af_{20,2}$	…	$af_{20,37}$	$af_{20,38}$

The opposition for a team in any given slot is represented by a positive integer number[11]. The negative values in the schedule (Table E1) are absoluted (i.e. $\sqrt{f^2}$).

$$af_{t,s} > 0, integer\ representing\ the\ opposition\ of\ team\ t\ in\ slot\ s$$

C102a. Avoid strength fixtures on the first game of the season.

$$if \sum_{\varepsilon=\varepsilon_1}^{|\varepsilon|} af_{\varepsilon,1} = team\ in\ set\ \varepsilon, then\ (count\ violations \div 2) \cdot \omega_{C102a} \qquad \forall\ \varepsilon$$

Note: Since the reverse fixtures are also counted in this sum function, the cumulated count is divided by two (i.e. if team t_1 plays team t_4 in slot s_1, then $af_{1,1}$ and $af_{4,1}$ represent the same single fixture but is counted twice in this formulation and therefore must be divided by two).

C102b. Avoid strength fixtures within the holiday period.

$$if \sum_{\varepsilon=\varepsilon_1}^{|\varepsilon|} af_{\varepsilon,\tau_1} + af_{\varepsilon,\tau_2} = team\ in\ set\ \varepsilon, then\ (count\ violations \div 2) \cdot \omega_{C102b} \qquad \forall\ \varepsilon$$

C102c. EPL matches prior to a UCL fixture should not be against strength group teams.

$$if \sum_{\phi_t=\phi_{t1}}^{|\phi_t|} af_{\phi_t,\phi_{fx}} = team\ in\ set\ \varepsilon, then\ count\ violations \cdot \omega_{C102c} \qquad \forall\ \phi_{fx}$$

C102d. EPL matches prior to a UEL fixture should not be against strength group teams.

$$if \sum_{\lambda_t=\lambda_{t1}}^{|\lambda_t|} af_{\lambda_t,\lambda_{fx}} = team\ in\ set\ \varepsilon, then\ count\ violations \cdot \omega_{C102d} \qquad \forall\ \lambda_{fx}$$

[11] Example: if $f_{19,2} = +5$, then $af_{19,2} = 5$ since team 19 has a fixture against team t_5 in slot s_2. Similarly if $f_{19,2} = -5$, then $af_{19,2} = 5$ since team 19 is still playing against team t_5 in slot s_2.

C102e. EPL matches succeeding a UCL fixture should not be against strength group teams.

$$if \sum_{\phi_t=\phi_{t1}}^{|\phi_t|} af_{\phi_t,\phi_{fy}} = team\ in\ set\ \varepsilon, then\ count\ violations \cdot \omega_{C102e} \qquad \forall\ \phi_{fx}$$

C102f. EPL matches succeeding a UEL fixture should not be against strength group teams.

$$if \sum_{\lambda_t=\lambda_{t1}}^{|\lambda_t|} af_{\lambda_t,\lambda_{fy}} = team\ in\ set\ \varepsilon, then\ count\ violations \cdot \omega_{C102f} \qquad \forall\ \lambda_{fy}$$

C108. The Boxing Day fixtures should not be repeated on New Year's Day.

$$if \sum_{t=1}^{20} bf_{t,\tau_1} - bf_{t,\tau_2} = 0, then\ (count\ violations \div 2) \cdot \omega_{C108}$$

C103. Teams in the paired group should not be assigned to holiday period fixtures.

$$if \sum_{\gamma_p=\gamma_{p1}}^{|\gamma_p|} bf_{\gamma_{px},\tau} = bf_{\gamma_{py},\tau}, then\ (count\ violations \div 2) \cdot \omega_{C103} \qquad \forall\ \tau$$

C104. The first round-robin must be completed before a fixture can be repeated in a 2RR.

$$if \sum_{s=1}^{19} bf_{t,s} - \sum_{s=20}^{38} bf_{t,s} \neq 0, then\ count\ violations \cdot \omega_{C104} \qquad \forall\ t$$

C112. The sum of each absolute entry in row *t* (team) in the timetable *T Set* of a compact 2RR consisting of *n* teams must equal *2(∑n)-2t*.

$$if \sum_{s=1}^{38} bf_{t,s} \neq 2\left(\sum_{n=1}^{20} n\right) - 2t, then\ count\ violations \cdot \omega_{C112} \qquad \forall\ t$$

C113. The sum of each absolute entry in column *s* (slot) in the timetable *T Set* consisting of *n* teams in a compact schedule must equal ∑*n*.

$$if \sum_{t=1}^{20} bf_{t,s} \neq \sum_{n=1}^{20} n, then\ count\ violations \cdot \omega_{C113} \qquad \forall\ s$$

C19. There should be at least $k = 12$ slots between two fixtures with the same opponents (i.e. there are 12 fixtures between Ω_1 and Ω_2).

$$if\ duplicate\ fixtures\ exist\ in\ bf_{t,\Omega_1} \rightarrow bf_{t,\Omega_2}, then\ count\ violations \cdot \omega_{C19} \qquad \forall\ t$$

C) Holiday period driving distance:

Since the distance parameter table was used in the simulated annealing solver as well as the pre-solver BIP, the pre-solver BIP distance table (Table D1) is repeated in Table E4.

Table E4: Driving distances parameter matrix (D)

i, j	t_1	t_2	t_3-t_{18}	t_{19}	t_{20}
t_1	$d_{1,1}$	$d_{1,2}$	…	$d_{1,19}$	$d_{1,20}$
t_2	$d_{2,1}$	$d_{2,2}$	…	$d_{2,19}$	$d_{2,20}$
t_3-t_{18}	⋮	⋮		⋮	⋮
t_{19}	$d_{19,1}$	$d_{19,2}$	…	$d_{19,19}$	$d_{19,20}$
t_{20}	$d_{20,1}$	$d_{20,2}$	…	$d_{20,19}$	$d_{20,20}$

$$d_{i,j} \geq 0$$

C107. The travelling distance to away holiday period fixtures should be minimised (i.e. compare the total holiday period distance of a candidate solution to the optimal distance determined using the pre-solver BIP formulated in Appendix D).

$$\sum_{t=1}^{20} d_{t,af_{t,\tau_1}} + \sum_{t=1}^{20} d_{t,af_{t,\tau_2}} - 2\delta = \textit{then count} \cdot \omega_{C107}$$

Note: Since the above function sums the distance of each fixture in each holiday slot twice (i.e. the reverse is calculated too), the δ distance, determined in the binary integer program, must be doubled.

APPENDIX F

POLYGON INITIAL TRIAL SOLUTION

The initial trial schedule required by the simulated annealing solver was constructed using the polygon construction principles described in Section 3.3.7 and is presented in matrix form here.

Table F1: Polygon schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	20	3	5	7	9	11	13	15	17	19	-2	-4	-6	-8	-10	-12	-14	-16	-18
t_2	-19	20	4	6	8	10	12	14	16	18	1	-3	-5	-7	-9	-11	-13	-15	-17
t_3	-18	-1	20	5	7	9	11	13	15	17	19	2	-4	-6	-8	-10	-12	-14	-16
t_4	-17	-19	-2	20	6	8	10	12	14	16	18	1	3	-5	-7	-9	-11	-13	-15
t_5	-16	-18	-1	-3	20	7	9	11	13	15	17	19	2	4	-6	-8	-10	-12	-14
t_6	-15	-17	-19	-2	-4	20	8	10	12	14	16	18	1	3	5	-7	-9	-11	-13
t_7	-14	-16	-18	-1	-3	-5	20	9	11	13	15	17	19	2	4	6	-8	-10	-12
t_8	-13	-15	-17	-19	-2	-4	-6	20	10	12	14	16	18	1	3	5	7	-9	-11
t_9	-12	-14	-16	-18	-1	-3	-5	-7	20	11	13	15	17	19	2	4	6	8	-10
t_{10}	-11	-13	-15	-17	-19	-2	-4	-6	-8	20	12	14	16	18	1	3	5	7	9
t_{11}	10	-12	-14	-16	-18	-1	-3	-5	-7	-9	20	13	15	17	19	2	4	6	8
t_{12}	9	11	-13	-15	-17	-19	-2	-4	-6	-8	-10	20	14	16	18	1	3	5	7
t_{13}	8	10	12	-14	-16	-18	-1	-3	-5	-7	-9	-11	20	15	17	19	2	4	6
t_{14}	7	9	11	13	-15	-17	-19	-2	-4	-6	-8	-10	-12	20	16	18	1	3	5
t_{15}	6	8	10	12	14	-16	-18	-1	-3	-5	-7	-9	-11	-13	20	17	19	2	4
t_{16}	5	7	9	11	13	15	-17	-19	-2	-4	-6	-8	-10	-12	-14	20	18	1	3
t_{17}	4	6	8	10	12	14	16	-18	-1	-3	-5	-7	-9	-11	-13	-15	20	19	2
t_{18}	3	5	7	9	11	13	15	17	-19	-2	-4	-6	-8	-10	-12	-14	-16	20	1
t_{19}	2	4	6	8	10	12	14	16	18	-1	-3	-5	-7	-9	-11	-13	-15	-17	20
t_{20}	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	-20	-3	-5	-7	-9	-11	-13	-15	-17	-19	2	4	6	8	10	12	14	16	18
t_2	19	-20	-4	-6	-8	-10	-12	-14	-16	-18	-1	3	5	7	9	11	13	15	17
t_3	18	1	-20	-5	-7	-9	-11	-13	-15	-17	-19	-2	4	6	8	10	12	14	16
t_4	17	19	2	-20	-6	-8	-10	-12	-14	-16	-18	-1	-3	5	7	9	11	13	15
t_5	16	18	1	3	-20	-7	-9	-11	-13	-15	-17	-19	-2	-4	6	8	10	12	14
t_6	15	17	19	2	4	-20	-8	-10	-12	-14	-16	-18	-1	-3	-5	7	9	11	13
t_7	14	16	18	1	3	5	-20	-9	-11	-13	-15	-17	-19	-2	-4	-6	8	10	12
t_8	13	15	17	19	2	4	6	-20	-10	-12	-14	-16	-18	-1	-3	-5	-7	9	11
t_9	12	14	16	18	1	3	5	7	-20	-11	-13	-15	-17	-19	-2	-4	-6	-8	10
t_{10}	11	13	15	17	19	2	4	6	8	-20	-12	-14	-16	-18	-1	-3	-5	-7	-9
t_{11}	-10	12	14	16	18	1	3	5	7	9	-20	-13	-15	-17	-19	-2	-4	-6	-8
t_{12}	-9	-11	13	15	17	19	2	4	6	8	10	-20	-14	-16	-18	-1	-3	-5	-7
t_{13}	-8	-10	-12	14	16	18	1	3	5	7	9	11	-20	-15	-17	-19	-2	-4	-6
t_{14}	-7	-9	-11	-13	15	17	19	2	4	6	8	10	12	-20	-16	-18	-1	-3	-5
t_{15}	-6	-8	-10	-12	-14	16	18	1	3	5	7	9	11	13	-20	-17	-19	-2	-4
t_{16}	-5	-7	-9	-11	-13	-15	17	19	2	4	6	8	10	12	14	-20	-18	-1	-3
t_{17}	-4	-6	-8	-10	-12	-14	-16	18	1	3	5	7	9	11	13	15	-20	-19	-2
t_{18}	-3	-5	-7	-9	-11	-13	-15	-17	19	2	4	6	8	10	12	14	16	-20	-1
t_{19}	-2	-4	-6	-8	-10	-12	-14	-16	-18	1	3	5	7	9	11	13	15	17	-20
t_{20}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

APPENDIX G
MATLAB CODE

The Matlab code of the EPL scheduling algorithm is approximately 2000 lines in length and too cumbersome to present here. An electronic copy of the code is available on the CD.

The pre-solver BIP code is named:

a. *EPLHolidayBIP_DanieleFerreira.m*

The two simulated annealing algorithms (hill-climbing algorithm included in each) can be found in files named:

b. *EPLSchedulingProblem_SimulatedAnnealing_DaVinci_DanieleFerreira.m*, and
c. *EPLSchedulingProblem_SimulatedAnnealing_Tesla_DanieleFerreira.m*.

For validation purposes, the problem parameters function for the 2011/12, 2012/13 and 2013/14 seasons are presented in a file named:

d. *problem_parameters.m*.

Note: since the Matlab waitbar GUI function consumes a lot of CPU time, it was commented-out for experimental purposes (i.e. observations and results). It is not however commented-out in the code placed on the CD.

APPENDIX H

OFFICIAL, PROPOSED AND BEST FOUND SCHEDULES

This appendix presents the official, proposed and best found EPL schedules for the 2013/14, 2012/13 and 2011/12 seasons. Table H10 presents the cost breakdown associated with each.

Official schedules

<u>2013/14</u>

Table H1: 2013/14 official schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	-9	3	-7	20	-2	8	-17	14	13	-12	4	-19	-5	6	16	-15	10	-18	-11
t_2	16	-19	18	-13	1	-15	6	-10	-3	11	-17	5	9	-8	-14	4	-12	7	20
t_3	18	-1	15	-6	12	-5	-11	19	2	-16	8	-10	14	-17	-13	20	-4	9	7
t_4	15	-12	5	-17	13	-9	-8	11	-20	7	-1	14	-19	18	6	-2	3	-10	-16
t_5	-20	9	-4	11	-19	3	10	-15	18	-6	16	-2	1	-12	-17	7	-14	8	13
t_6	-11	8	-19	3	-10	16	-2	18	-15	5	-20	7	13	-1	-4	12	-9	17	14
t_7	13	-15	1	-9	14	-17	20	-16	8	-4	12	-6	-18	11	10	-5	19	-2	-3
t_8	14	-6	9	-12	17	-1	4	-13	-7	20	-3	15	-16	2	11	-19	18	-5	-10
t_9	1	-5	-8	7	-20	4	-14	17	10	-19	13	-12	-2	16	18	-11	6	-3	-15
t_{10}	19	-16	13	-14	6	-18	-5	2	-9	15	-11	3	12	-20	-7	17	-1	4	8
t_{11}	6	-18	14	-5	15	-13	3	-4	19	-2	10	-16	20	-7	-8	9	-17	12	1
t_{12}	-17	4	-16	8	-3	19	13	-20	-14	1	-7	9	-10	5	15	-6	2	-11	-18
t_{13}	-7	20	-10	2	-4	11	-12	8	-1	14	-9	17	-6	19	3	-18	15	-16	-5
t_{14}	-8	17	-11	10	-7	20	9	-1	12	-13	18	-4	-3	15	2	-16	5	-19	-6
t_{15}	-4	7	-3	16	-11	2	-18	5	6	-10	19	-8	17	-14	-12	1	-13	20	9
t_{16}	-2	10	12	-15	18	-6	-19	7	-17	3	-5	11	8	-9	-1	14	-20	13	4
t_{17}	12	-14	-20	4	-8	7	1	-9	16	-18	2	-13	-15	3	5	-10	11	-6	-19
t_{18}	-3	11	-2	19	-16	10	15	-6	-5	17	-14	20	7	-4	-9	13	-8	1	12
t_{19}	-10	2	6	-18	5	-12	16	-3	-11	9	-15	1	4	-13	-20	8	-7	14	17
t_{20}	5	-13	17	-1	9	-14	-7	12	4	-8	6	-18	-11	10	19	-3	16	-15	-2

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	5	9	-3	19	-13	12	-4	-20	2	-8	7	-10	15	-16	18	-6	11	17	-14
t_2	-9	-16	19	-5	3	-11	17	13	-1	15	-18	12	-4	14	-7	8	-20	-6	10
t_3	-14	-18	1	10	-2	16	-8	6	-12	5	-15	4	-20	13	-9	17	-7	11	-19
t_4	19	-15	12	-14	20	-7	1	17	-13	9	-5	-3	2	-6	10	-18	16	8	-11
t_5	-1	20	-9	2	-18	6	-16	-11	19	-3	4	14	-7	17	-8	12	-13	-10	15
t_6	-13	11	-8	-7	15	-5	20	-3	10	-16	19	9	-12	4	-17	1	-14	2	-18
t_7	18	-13	15	6	-8	4	-12	9	-14	17	-1	-19	5	-10	2	-11	3	-20	16
t_8	16	-14	6	-15	7	-20	3	12	-17	1	-9	-18	19	-11	5	-2	10	-4	13
t_9	2	-1	5	12	-10	19	-13	-7	20	-4	8	-6	11	-18	3	-16	15	14	-17
t_{10}	-12	-19	16	-3	9	-15	11	14	-6	18	-13	1	-17	7	-4	20	-8	5	-2
t_{11}	-20	-6	18	16	-19	2	-10	5	-15	13	-14	17	-9	8	-12	7	-1	-3	4
t_{12}	10	17	-4	-9	14	-1	7	-8	3	-19	16	-2	6	-15	11	-5	18	-13	20
t_{13}	6	7	-20	-17	1	-14	9	-2	4	-11	10	-15	18	-3	16	-19	5	12	-8
t_{14}	3	8	-17	4	-12	13	-18	-10	7	-20	11	-5	16	-2	19	-15	6	-9	1
t_{15}	-17	4	-7	8	-6	10	-19	-16	11	-2	3	13	-1	12	-20	14	-9	18	-5
t_{16}	-8	2	-10	-11	17	-3	5	15	-18	6	-12	20	-14	1	-13	9	-4	19	-7
t_{17}	15	-12	14	13	-16	18	-2	-4	8	-7	20	-11	10	-5	6	-3	19	-1	9
t_{18}	-7	3	-11	-20	5	-17	14	-19	16	-10	2	8	-13	9	-1	4	-12	-15	6
t_{19}	-4	10	-2	-1	11	-9	15	18	-5	12	-6	7	-8	20	-14	13	-17	-16	3
t_{20}	11	-5	13	18	-4	8	-6	1	-9	14	-17	-16	3	-19	15	-10	2	7	-12

2012/13

Table H2: 2012/13 official schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	20	-8	17	-14	3	-9	13	-10	11	-18	4	16	-6	-15	7	2	-5	19	-13
t_2	-7	9	-20	15	-8	4	-5	14	-6	3	-16	-12	17	18	-19	-1	13	-11	5
t_3	13	-14	-8	20	-1	6	-18	-12	17	-2	9	4	-16	-7	11	10	-19	-15	18
t_4	-5	10	12	-19	17	-2	16	6	-20	15	-1	-3	18	8	-9	-7	11	14	-16
t_5	4	-6	16	-7	12	-19	2	-13	10	-8	18	11	-20	-14	15	-9	1	17	-2
t_6	-15	5	19	-17	14	-3	12	-4	2	-11	8	-10	1	9	-18	-13	20	16	-12
t_7	2	-16	-10	5	-11	20	-15	-17	8	-9	13	-19	12	3	-1	4	-14	-18	15
t_8	-10	1	3	-13	2	-12	14	19	-7	5	-6	15	-11	-4	20	-18	16	9	-14
t_9	12	-2	-18	10	-15	1	-20	16	-19	7	-3	13	-14	-6	4	5	-17	-8	20
t_{10}	8	-4	7	-9	19	-16	17	1	-5	20	-15	6	-13	-11	14	-3	18	12	-17
t_{11}	-17	18	13	-16	7	-14	19	15	-1	6	-20	-5	8	10	-3	12	-4	2	-19
t_{12}	-9	17	-4	18	-5	8	-6	3	-16	14	-19	2	-7	-20	13	-11	15	-10	6
t_{13}	-3	19	-11	8	-18	15	-1	5	-14	16	-7	-9	10	17	-12	6	-2	-20	1
t_{14}	-19	3	-15	1	-6	11	-8	-2	13	-12	17	-18	9	5	-10	-16	7	-4	8
t_{15}	6	-20	14	-2	9	-13	7	-11	18	-4	10	-8	19	1	-5	17	-12	3	-7
t_{16}	-18	7	-5	11	-20	10	-4	-9	12	-13	2	-1	3	19	-17	14	-8	-6	4
t_{17}	11	-12	-1	6	-4	18	-10	7	-3	19	-14	20	-2	-13	16	-15	9	-5	10
t_{18}	16	-11	9	-12	13	-17	3	20	-15	1	-5	14	-4	-2	6	8	-10	7	-3
t_{19}	14	-13	-6	4	-10	5	-11	-8	9	-17	12	7	-15	-16	2	-20	3	-1	11
t_{20}	-1	15	2	-3	16	-7	9	-18	4	-10	11	-17	5	12	-8	19	-6	13	-9

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	-12	14	-3	9	-17	8	-20	6	-16	15	-7	5	-2	10	-4	18	-11	-19	12
t_2	10	-15	8	-4	20	-9	7	-17	12	-18	19	-13	1	-14	16	-3	6	11	-10
t_3	5	-20	1	-6	8	14	-13	16	-4	7	-11	19	-10	12	-9	2	-17	15	-5
t_4	-13	19	-17	2	-12	-10	5	-18	3	-8	9	-11	7	-6	1	-15	20	-14	13
t_5	-3	7	-12	19	-16	6	-4	20	-11	14	-15	-1	9	13	-10	8	-18	-17	3
t_6	-7	17	-14	3	-19	-5	15	-1	10	-9	18	-20	13	4	-8	11	-2	-16	7
t_7	6	-5	11	-20	10	16	-2	-12	19	-3	1	14	-4	17	-13	9	-8	18	-6
t_8	-17	13	-2	12	-3	-1	10	11	-15	4	-20	-16	18	-19	6	-5	7	-9	17
t_9	11	-10	15	-1	18	2	-12	14	-13	6	-4	17	-5	-16	3	-7	19	8	-11
t_{10}	-2	9	-19	16	-7	4	-8	13	-6	11	-14	-18	3	-1	5	-20	15	-12	2
t_{11}	-9	16	-7	14	-13	-18	17	-8	5	-10	3	4	-12	-15	20	-6	1	-2	9
t_{12}	1	-18	5	-8	4	-17	9	7	-2	20	-13	-15	11	-3	19	-14	16	10	-1
t_{13}	4	-8	18	-15	11	-19	3	-10	9	-17	12	2	-6	-5	7	-16	14	20	-4
t_{14}	20	-1	6	-11	15	-3	19	-9	18	-5	10	-7	16	2	-17	12	-13	4	-20
t_{15}	-16	2	-9	13	-14	20	-6	-19	8	-1	5	12	-17	11	-18	4	-10	-3	16
t_{16}	15	-11	20	-10	5	-7	18	-3	1	-19	17	8	-14	9	-2	13	-12	6	-15
t_{17}	8	-6	4	-18	1	12	-11	2	-20	13	-16	-9	15	-7	14	-19	3	5	-8
t_{18}	-19	12	-13	17	-9	11	-16	4	-14	2	-6	10	-8	-20	15	-1	5	-7	19
t_{19}	18	-4	10	-5	6	13	-14	15	-7	16	-2	-3	20	8	-12	17	-9	1	-18
t_{20}	-14	3	-16	7	-2	-15	1	-5	17	-12	8	6	-19	18	-11	10	-4	-13	14

2011/12

Table H3: 2011/12 official schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	-11	5	4	-14	2	-13	18	-6	3	-7	10	-20	12	-9	17	-19	-8	16	15
t_2	-13	11	18	-10	-1	20	-14	7	-19	4	-15	6	17	-12	3	-16	-5	8	9
t_3	20	-14	-5	16	-8	7	-15	9	-1	17	-19	12	-6	18	-2	4	13	-11	-10
t_4	-12	6	-1	20	-15	14	-5	10	13	-2	11	-18	8	-16	7	-3	-9	17	19
t_5	7	-1	3	-17	6	-16	4	-12	-15	19	-8	9	-11	14	-13	10	2	-18	-20
t_6	10	-4	14	-13	-5	17	-7	1	18	-11	20	-2	3	-8	19	-9	-16	15	12
t_7	-5	19	-15	9	16	-3	6	-2	-8	1	-12	17	-14	13	-4	18	20	-10	-11
t_8	9	-17	-12	15	3	-11	19	-13	7	-16	5	-10	-4	6	-20	14	1	-2	-18
t_9	-8	15	17	-7	12	-19	16	-3	11	-10	18	-5	-20	1	-14	6	4	-13	-2
t_{10}	-6	12	-20	2	13	-18	11	-4	-14	9	-1	8	16	-17	15	-5	-19	7	3
t_{11}	1	-2	13	-18	-20	8	-10	17	-9	6	-4	14	5	-19	16	-15	-12	3	7
t_{12}	4	-10	8	-19	-9	15	-17	5	16	-13	7	-3	-1	2	-18	20	11	-14	-6
t_{13}	2	-18	-11	6	-10	1	-20	8	-4	12	-14	19	15	-7	5	-17	-3	9	16
t_{14}	-19	3	-6	1	18	-4	2	-16	10	-20	13	-11	7	-5	9	-8	-15	12	17
t_{15}	17	-9	7	-8	4	-12	3	-19	5	-18	2	-16	-13	20	-10	11	14	-6	-1
t_{16}	18	-20	19	-3	-7	5	-9	14	-12	8	-17	15	-10	4	-11	2	6	-1	-13
t_{17}	-15	8	-9	5	19	-6	12	-11	20	-3	16	-7	-2	10	-1	13	18	-4	-14
t_{18}	-16	13	-2	11	-14	10	-1	20	-6	15	-9	4	19	-3	12	-7	-17	5	8
t_{19}	14	-7	-16	12	-17	9	-8	15	2	-5	3	-13	-18	11	-6	1	10	-20	-4
t_{20}	-3	16	10	-4	11	-2	13	-18	-17	14	-6	1	9	-15	8	-12	-7	19	5

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	-12	14	-4	13	-2	6	-18	-5	11	-17	8	-15	19	-16	9	7	-3	20	-10
t_2	-17	10	-18	-20	1	-7	14	-11	13	-3	5	-9	16	-8	12	-4	19	-6	15
t_3	6	-16	5	-7	8	-9	15	14	-20	2	-13	10	-4	11	-18	-17	1	-12	19
t_4	-8	-20	1	-14	15	-10	5	-6	12	-7	9	-19	3	-17	16	2	-13	18	-11
t_5	11	17	-3	16	-6	12	-4	1	-7	13	-2	20	-10	18	-14	-19	15	-9	8
t_6	-3	13	-14	-17	5	-1	7	4	-10	-19	16	-12	9	-15	8	11	-18	2	-20
t_7	14	-9	15	3	-16	2	-6	-19	5	4	-20	11	-18	10	-13	-1	8	-17	12
t_8	4	-15	12	11	-3	13	-19	17	-9	20	-1	18	-14	2	-6	16	-7	10	-5
t_9	20	7	-17	19	-12	3	-16	-15	8	14	-4	2	-6	13	-1	10	-11	5	-18
t_{10}	-16	-2	20	18	-13	4	-11	-12	6	-15	19	-3	5	-7	17	-9	14	-8	1
t_{11}	-5	18	-13	-8	20	-17	10	2	-1	-16	12	-7	15	-3	19	-6	9	-14	4
t_{12}	1	19	-8	-15	9	-5	17	10	-4	18	-11	6	-20	14	-2	13	-16	3	-7
t_{13}	-15	-6	11	-1	10	-8	20	18	-2	-5	3	-16	17	-9	7	-12	4	-19	14
t_{14}	-7	-1	6	4	-18	16	-2	-3	19	-9	15	-17	8	-12	5	20	-10	11	-13
t_{15}	13	8	-7	12	-4	19	-3	9	-17	10	-14	1	-11	6	-20	18	-5	16	-2
t_{16}	10	3	-19	-5	7	-14	9	20	-18	11	-6	13	-2	1	-4	-8	12	-15	17
t_{17}	2	-5	9	6	-19	11	-12	-8	15	1	-18	14	-13	4	-10	3	-20	7	-16
t_{18}	-19	-11	2	-10	14	-20	1	-13	16	-12	17	-8	7	-5	3	-15	6	-4	9
t_{19}	18	-12	16	-9	17	-15	8	7	-14	6	-10	4	-1	20	-11	5	-2	13	-3
t_{20}	-9	4	-10	2	-11	18	-13	-16	3	-8	7	-5	12	-19	15	-14	17	-1	6

Proposed schedules

2013/14

Table H4: 2013/14 proposed schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	16	-2	-9	8	-3	-20	10	-18	17	13	-5	12	-15	6	-14	19	-4	7	11
t_2	-17	1	6	-15	11	19	-20	8	-13	-18	4	-5	14	-7	12	-9	3	-16	-10
t_3	13	-10	12	-9	1	15	-8	-19	11	-20	17	16	-6	5	7	-14	-2	4	18
t_4	-11	20	-5	6	-18	-14	15	9	-10	19	-2	-17	7	13	-16	-12	1	-3	-8
t_5	10	-19	4	-7	8	12	-14	-6	20	-9	1	2	-16	-3	17	13	-18	11	15
t_6	-8	14	-2	-4	9	-16	-7	5	15	-12	20	-10	3	-1	11	-17	19	18	-13
t_7	18	-15	17	5	-19	13	6	-12	-8	14	-10	11	-4	2	-3	16	-20	-1	9
t_8	6	-16	20	-1	-5	11	3	-2	7	-17	14	-15	-18	19	-13	10	-12	9	4
t_9	15	-12	1	3	-6	17	16	-4	-14	5	-19	20	-11	18	-10	2	13	-8	-7
t_{10}	-5	3	15	-20	17	18	-1	-13	4	-11	7	6	-19	14	9	-8	-16	12	2
t_{11}	4	-13	-14	19	-2	-8	18	20	-3	10	-16	-7	9	12	-6	15	17	-5	-1
t_{12}	-20	9	-3	16	-15	-5	13	7	-19	6	-18	-1	17	-11	-2	4	8	-10	-14
t_{13}	-3	11	-16	18	14	-7	-12	10	2	-1	15	19	-20	-4	8	-5	-9	17	6
t_{14}	19	-6	11	-17	-13	4	5	-16	9	-7	-8	18	-2	-10	1	3	-15	20	12
t_{15}	-9	7	-10	2	12	-3	-4	17	-6	16	-13	8	1	-20	18	-11	14	-19	-5
t_{16}	-1	8	13	-12	20	6	-9	14	18	-15	11	-3	5	-17	4	-7	10	2	-19
t_{17}	2	-18	-7	14	-10	-9	19	-15	-1	8	-3	4	-12	16	-5	6	-11	-13	20
t_{18}	-7	17	19	-13	4	-10	-11	1	-16	2	12	-14	8	-9	-15	20	5	-6	-3
t_{19}	-14	5	-18	-11	7	-2	-17	3	12	-4	9	-13	10	-8	20	-1	-6	15	16
t_{20}	12	-4	-8	10	-16	1	2	-11	-5	3	-6	-9	13	15	-19	-18	7	-14	-17

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	-17	-8	9	3	-10	2	-16	20	18	-6	14	-13	5	-12	15	-7	-19	4	-11
t_2	13	15	-6	-11	20	-1	17	-19	-8	7	-12	18	-4	5	-14	16	9	-3	10
t_3	-11	9	-12	-1	8	10	-13	-15	19	-5	-7	20	-17	-16	6	-4	14	2	-18
t_4	10	-6	5	18	-15	-20	11	14	-9	-13	16	-19	2	17	-7	3	12	-1	8
t_5	-20	7	-4	-8	14	19	-10	-12	6	3	-17	9	-1	-2	16	-11	-13	18	-15
t_6	-15	4	2	-9	7	-14	8	16	-5	1	-11	12	-20	10	-3	-18	17	-19	13
t_7	8	-5	-17	19	-6	15	-18	-13	12	-2	3	-14	10	-11	4	1	-16	20	-9
t_8	-7	1	-20	5	-3	16	-6	-11	2	-19	13	17	-14	15	18	-9	-10	12	-4
t_9	14	-3	-1	6	-16	12	-15	-17	4	-18	10	-5	19	-20	11	8	-2	-13	7
t_{10}	-4	20	-15	-17	1	-3	5	-18	13	-14	-9	11	-7	-6	19	-12	8	16	-2
t_{11}	3	-19	14	2	-18	13	-4	8	-20	-12	6	-10	16	7	-9	5	-15	-17	1
t_{12}	19	-16	3	15	-13	-9	20	5	-7	11	2	-6	18	1	-17	10	-4	-8	14
t_{13}	-2	-18	16	-14	12	-11	3	7	-10	4	-8	1	-15	-19	20	-17	5	9	-6
t_{14}	-9	17	-11	13	-5	6	-19	-4	16	10	-1	7	8	-18	2	-20	-3	15	-12
t_{15}	6	-2	10	-12	4	-7	9	3	-17	20	-18	-16	13	-8	-1	19	11	-14	5
t_{16}	-18	12	-13	-20	9	-8	1	-6	-14	17	-4	15	-11	3	-5	-2	7	-10	19
t_{17}	1	-14	7	10	-19	18	-2	9	15	-16	5	-8	3	-4	12	13	-6	11	-20
t_{18}	16	13	-19	-4	11	-17	7	10	-1	9	15	-2	-12	14	-8	6	-20	-5	3
t_{19}	-12	11	18	-7	17	-5	14	2	-3	8	-20	4	-9	13	-10	-15	1	6	-16
t_{20}	5	-10	8	16	-2	4	-12	-1	11	-15	19	-3	6	9	-13	14	18	-7	17

2012/13

Table H5: 2012/13 proposed schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	6	-3	-17	11	15	-16	4	-5	8	-9	2	-13	7	-10	-20	18	14	-19	12
t_2	-17	4	3	-12	-5	18	-19	13	-15	6	-1	10	-8	16	14	-9	-11	20	-7
t_3	-8	1	-2	18	-4	-11	5	-19	-17	7	-15	-20	6	-14	10	-12	16	13	-9
t_4	7	-2	8	-16	3	14	-1	15	6	-12	17	19	-9	20	-13	11	-10	-5	18
t_5	-9	17	6	-14	2	10	-3	1	-7	-18	8	-15	-12	13	-19	-16	20	4	-11
t_6	-1	13	-5	15	20	-7	10	14	-4	-2	19	-11	-3	12	-18	-8	9	16	-17
t_7	-4	20	19	-8	-10	6	-14	16	5	-3	13	18	-1	9	12	-17	-15	11	2
t_8	3	-19	-4	7	13	-9	20	-10	-1	17	-5	-16	2	-18	-11	6	12	-14	-15
t_9	5	-10	13	-17	-14	8	-16	-11	19	1	-20	12	4	-7	15	2	-6	-18	3
t_{10}	-16	9	18	-19	7	-5	-6	8	-11	15	12	-2	-14	1	-3	-13	4	17	-20
t_{11}	-20	15	14	-1	-18	3	-12	9	10	-19	16	6	-13	17	8	-4	2	-7	5
t_{12}	19	-14	-20	2	16	-17	11	-18	-13	4	-10	-9	5	-6	-7	3	-8	15	-1
t_{13}	18	-6	-9	20	-8	-15	17	-2	12	16	-7	1	11	-5	4	10	-19	-3	14
t_{14}	-15	12	-11	5	9	-4	7	-6	-16	20	-18	-17	10	3	-2	19	-1	8	-13
t_{15}	14	-11	16	-6	-1	13	-18	-4	2	-10	3	5	-17	19	-9	-20	7	-12	8
t_{16}	10	-18	-15	4	-12	1	9	-7	14	-13	-11	8	20	-2	17	5	-3	-6	19
t_{17}	2	-5	1	9	-19	12	-13	-20	3	-8	-4	14	15	-11	-16	7	-18	-10	6
t_{18}	-13	16	-10	-3	11	-2	15	12	-20	5	14	-7	-19	8	6	-1	17	9	-4
t_{19}	-12	8	-7	10	17	-20	2	3	-9	11	-6	-4	18	-15	5	-14	13	1	-16
t_{20}	11	-7	12	-13	-6	19	-8	17	18	-14	9	3	-16	-4	1	15	-5	-2	10

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	5	-8	-4	17	-7	-2	16	-11	19	-12	9	-15	13	-14	-18	10	20	-6	3
t_2	-13	15	19	-3	8	1	-18	12	-20	7	-6	5	-10	11	9	-16	-14	17	-4
t_3	19	17	-5	2	-6	15	11	-18	-13	9	-7	4	20	-16	12	14	-10	8	-1
t_4	-15	-6	1	-8	9	-17	-14	16	5	-18	12	-3	-19	10	-11	-20	13	-7	2
t_5	-1	7	3	-6	12	-8	-10	14	-4	11	18	-2	15	-20	16	-13	19	9	-17
t_6	-14	4	-10	5	3	-19	7	-15	-16	17	2	-20	11	-9	8	-12	18	1	-13
t_7	-16	-5	14	-19	1	-13	-6	8	-11	-2	3	10	-18	15	17	-9	-12	4	-20
t_8	10	1	-20	4	-2	5	9	-7	14	15	-17	-13	16	-12	-6	18	11	-3	19
t_9	11	-19	16	-13	-4	20	-8	17	18	-3	-1	14	-12	6	-2	7	-15	-5	10
t_{10}	-8	11	6	-18	14	-12	5	19	-17	20	-15	-7	2	-4	13	-1	3	16	-9
t_{11}	-9	-10	12	-14	13	-16	-3	1	7	-5	19	18	-6	-2	4	-17	-8	20	-15
t_{12}	18	13	-11	20	-5	10	17	-2	-15	1	-4	-16	9	8	-3	6	7	-19	14
t_{13}	2	-12	-17	9	-11	7	15	-20	3	-14	-16	8	-1	19	-10	5	-4	-18	6
t_{14}	6	16	-7	11	-10	18	4	-5	-8	13	-20	-9	17	1	-19	-3	2	15	-12
t_{15}	4	-2	18	-16	17	-3	-13	6	12	-8	10	1	-5	-7	20	-19	9	-14	11
t_{16}	7	-14	-9	15	-20	11	-1	-4	6	-19	13	12	-8	3	-5	2	-17	-10	18
t_{17}	20	-3	13	-1	-15	4	-12	-9	10	-6	8	19	-14	18	-7	11	16	-2	5
t_{18}	-12	20	-15	10	19	-14	2	3	-9	4	-5	-11	7	-17	1	-8	-6	13	-16
t_{19}	-3	9	-2	7	-18	6	20	-10	-1	16	-11	-17	4	-13	14	15	-5	12	-8
t_{20}	-17	-18	8	-12	16	-9	-19	13	2	-10	14	6	-3	5	-15	4	-1	-11	7

2011/12

Table H6: 2011/12 proposed schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	-16	9	-12	-19	10	-7	-4	6	-3	18	-15	-11	5	13	-14	17	2	-20	8
t_2	-8	5	-13	15	-6	-9	12	11	-14	10	-7	19	17	-16	20	3	-1	4	18
t_3	19	-6	-5	8	-4	10	9	-12	1	-20	18	13	-11	-17	15	-2	-16	7	-14
t_4	-9	8	-15	12	3	-16	1	14	-6	17	-13	-20	18	-7	-11	10	-19	-2	5
t_5	20	-2	3	-10	-13	11	-17	-16	7	-12	6	8	-1	14	9	-15	18	19	-4
t_6	-10	3	-8	9	2	-17	16	-1	4	-11	-5	15	-14	-18	12	-20	-7	13	19
t_7	12	-15	20	-11	-8	1	-14	18	-5	16	2	-10	19	4	-17	9	6	-3	-13
t_8	2	-4	6	-3	7	20	-10	-9	16	-15	14	-5	-12	11	-19	-13	17	-18	-1
t_9	4	-1	14	-6	-16	2	-3	8	19	-13	11	-18	-15	20	-5	-7	10	-17	-12
t_{10}	6	-14	-18	5	-1	-3	8	-15	12	-2	17	7	-20	-19	13	-4	-9	16	-11
t_{11}	-18	17	-16	7	19	-5	13	-2	-20	6	-9	1	3	-8	4	14	-15	12	10
t_{12}	-7	16	1	-4	17	13	-2	3	-10	5	-19	-14	8	15	-6	18	20	-11	9
t_{13}	15	-19	2	-20	5	-12	-11	17	-18	9	4	-3	16	-1	-10	8	14	-6	7
t_{14}	-17	10	-9	16	-20	-18	7	-4	2	19	-8	12	6	-5	1	-11	-13	15	3
t_{15}	-13	7	4	-2	18	-19	-20	10	-17	8	1	-6	9	-12	-3	5	11	-14	16
t_{16}	1	-12	11	-14	9	4	-6	5	-8	-7	20	-17	-13	2	-18	19	3	-10	-15
t_{17}	14	-11	19	-18	-12	6	5	-13	15	-4	-10	16	-2	3	7	-1	-8	9	-20
t_{18}	11	-20	10	17	-15	14	19	-7	13	-1	-3	9	-4	6	16	-12	-5	8	-2
t_{19}	-3	13	-17	1	-11	15	-18	20	-9	-14	12	-2	-7	10	8	-16	4	-5	-6
t_{20}	-5	18	-7	13	14	-8	15	-19	11	3	-16	4	10	-9	-2	6	-12	1	17

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	11	-10	-6	3	-5	15	20	-8	-13	16	-9	14	12	-17	19	7	-18	4	-2
t_2	-19	6	-11	14	-17	7	-4	-18	16	8	-5	-20	13	-3	-15	9	-10	-12	1
t_3	-13	4	12	-1	11	-18	-7	14	17	-19	6	-15	5	2	-8	-10	20	-9	16
t_4	20	-3	-14	6	-18	13	2	-5	7	9	-8	11	15	-10	-12	16	-17	-1	19
t_5	-8	13	16	-7	1	-6	-19	4	-14	-20	2	-9	-3	15	10	-11	12	17	-18
t_6	-15	-2	1	-4	14	5	-13	-19	18	10	-3	-12	8	20	-9	17	11	-16	7
t_7	10	8	-18	5	-19	-2	3	13	-4	-12	15	17	-20	-9	11	-1	-16	14	-6
t_8	5	-7	9	-16	12	-14	18	1	-11	-2	4	19	-6	13	3	-20	15	10	-17
t_9	18	16	-8	-19	15	-11	17	12	-20	-4	1	5	-14	7	6	-2	13	3	-10
t_{10}	-7	1	15	-12	20	-17	-16	11	19	-6	14	-13	18	4	-5	3	2	-8	9
t_{11}	-1	-19	2	20	-3	9	-12	-10	8	18	-17	-4	16	-14	-7	5	-6	-13	15
t_{12}	14	-17	-3	10	-8	19	11	-9	-15	7	-16	6	-1	-18	4	-13	-5	2	-20
t_{13}	3	-5	-17	18	-16	-4	6	-7	1	-15	19	10	-2	-8	20	12	-9	11	-14
t_{14}	-12	20	4	-2	-6	8	-15	-3	5	17	-10	-1	9	11	-16	18	-19	-7	13
t_{15}	6	-18	-10	17	-9	-1	14	-16	12	13	-7	3	-4	-5	2	19	-8	20	-11
t_{16}	17	-9	-5	8	13	-20	10	15	-2	-1	12	18	-11	-19	14	-4	7	6	-3
t_{17}	-16	12	13	-15	2	10	-9	20	-3	-14	11	-7	-19	1	18	-6	4	-5	8
t_{18}	-9	15	7	-13	4	3	-8	2	-6	-11	20	-16	-10	12	-17	-14	1	-19	5
t_{19}	2	11	-20	9	7	-12	5	6	-10	3	-13	-8	17	16	-1	-15	14	18	-4
t_{20}	-4	-14	19	-11	-10	16	-1	-17	9	5	-18	2	7	-6	-13	8	-3	-15	12

Best schedules found

<u>2013/14</u>

Table H7: 2013/14 best found schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	-16	12	5	-7	4	13	-6	11	9	-14	20	8	-3	2	-18	-10	19	-15	-17
t_2	14	-17	-4	13	-3	-15	7	-10	-12	11	-5	-19	18	-1	6	20	-9	8	16
t_3	-7	18	-9	-14	2	-11	16	15	-4	-13	19	-20	1	-12	-17	8	5	-10	6
t_4	13	-6	2	11	-1	10	-14	-8	3	15	-9	5	-17	18	16	-19	12	20	-7
t_5	-15	7	-1	-10	17	-20	11	19	-18	-8	2	-4	16	-6	-14	9	-3	-12	13
t_6	-18	4	8	-17	19	-16	1	7	-20	-12	15	-11	9	5	-2	-13	10	-14	-3
t_7	3	-5	-15	1	-8	17	-2	-6	10	18	-13	14	-19	-20	9	12	-11	16	4
t_8	-20	11	-6	-19	7	-9	12	4	-16	5	18	-1	-13	14	-15	-3	17	-2	-10
t_9	-11	16	3	-15	18	8	-13	20	-1	-10	4	-12	-6	17	-7	-5	2	-19	-14
t_{10}	-19	15	-16	5	-14	-4	20	2	-7	9	17	-18	11	-13	-12	1	-6	3	8
t_{11}	9	-8	14	-4	12	3	-5	-1	13	-2	-16	6	-10	15	20	-17	7	-18	-19
t_{12}	17	-1	13	20	-11	19	-8	-16	2	6	-14	9	-15	3	10	-7	-4	5	-18
t_{13}	-4	20	-12	-2	15	-1	9	-18	-11	3	7	-16	8	10	-19	6	14	-17	-5
t_{14}	-2	19	-11	3	10	-18	4	17	-15	1	12	-7	20	-8	5	16	-13	6	9
t_{15}	5	-10	7	9	-13	2	-19	-3	14	-4	-6	17	12	-11	8	18	-16	1	20
t_{16}	1	-9	10	-18	20	6	-3	12	8	-17	11	13	-5	19	-4	-14	15	-7	-2
t_{17}	-12	2	-20	6	-5	-7	18	-14	-19	16	-10	-15	4	-9	3	11	-8	13	1
t_{18}	6	-3	19	16	-9	14	-17	13	5	-7	-8	10	-2	-4	1	-15	-20	11	12
t_{19}	10	-14	-18	8	-6	-12	15	-5	17	20	-3	2	7	-16	13	4	-1	9	11
t_{20}	8	-13	17	-12	-16	5	-10	-9	6	-19	-1	3	-14	7	-11	-2	18	-4	-15

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	7	-4	-9	3	-20	-8	14	-11	-13	6	16	-5	-12	18	-19	10	15	-2	17
t_2	-13	3	12	-18	5	19	-11	10	15	-7	-14	4	17	-6	9	-20	-8	1	-16
t_3	14	-2	4	-1	-19	20	13	-15	11	-16	7	9	-18	17	-5	-8	10	12	-6
t_4	-11	1	-3	17	9	-5	-15	8	-10	14	-13	-2	6	-16	-12	19	-20	-18	7
t_5	10	-17	18	-16	-2	4	8	-19	20	-11	15	1	-7	14	3	-9	12	6	-13
t_6	17	-19	20	-9	-15	11	12	-7	16	-1	18	-8	-4	2	-10	13	14	-5	3
t_7	-1	8	-10	19	13	-14	-18	6	-17	2	-3	15	5	-9	11	-12	-16	20	-4
t_8	19	-7	16	13	-18	1	-5	-4	9	-12	20	6	-11	15	-17	3	2	-14	10
t_9	15	-18	1	6	-4	12	10	-20	-8	13	11	-3	-16	7	-2	5	19	-17	14
t_{10}	-5	14	7	-11	-17	18	-9	-2	4	-20	19	16	-15	12	6	-1	-3	13	-8
t_{11}	4	-12	-13	10	16	-6	2	1	-3	5	-9	-14	8	-20	-7	17	18	-15	19
t_{12}	-20	11	-2	15	14	-9	-6	16	-19	8	-17	-13	1	-10	4	7	-5	-3	18
t_{13}	2	-15	11	-8	-7	16	-3	18	1	-9	4	12	-20	19	-14	-6	17	-10	5
t_{14}	-3	-10	15	-20	-12	7	-1	-17	18	-4	2	11	-19	-5	13	-16	-6	8	-9
t_{15}	-9	13	-14	-12	6	-17	4	3	-2	19	-5	-7	10	-8	16	-18	-1	11	-20
t_{16}	18	-20	-8	5	-11	-13	17	-12	-6	3	-1	-10	9	4	-15	14	7	-19	2
t_{17}	-6	5	19	-4	10	15	-16	14	7	-18	12	20	-2	-3	8	-11	-13	9	-1
t_{18}	-16	9	-5	2	8	-10	7	-13	-14	17	-6	-19	3	-1	20	15	-11	4	-12
t_{19}	-8	6	-17	-7	3	-2	-20	5	12	-15	-10	18	14	-13	1	-4	-9	16	-11
t_{20}	12	16	-6	14	1	-3	19	9	-5	10	-8	-17	13	11	-18	2	4	-7	15

2012/13

Table H8: 2012/13 best found schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	11	-14	-2	13	3	-15	-5	16	-18	20	-8	-4	12	-9	-17	19	-10	-6	7
t_2	-20	17	1	-8	-9	5	13	-10	11	-14	7	19	-18	6	15	-12	3	4	-16
t_3	10	-9	6	18	-1	-19	12	-14	16	-17	11	-13	7	-15	4	-8	-2	5	20
t_4	-8	16	-10	-17	20	9	-6	18	-13	7	-19	1	-5	14	-3	15	11	-2	-12
t_5	12	-11	-20	15	16	-2	1	8	-19	18	-17	-9	4	-10	-14	6	-7	-3	13
t_6	7	-10	-3	19	-14	17	4	-11	8	-16	12	-15	13	-2	9	-5	-20	1	18
t_7	-6	19	12	-14	13	11	-20	15	9	-4	-2	16	-3	8	18	-10	5	17	-1
t_8	4	-12	-18	2	-17	-20	14	-5	-6	19	1	-10	9	-7	-11	3	-13	-16	15
t_9	-16	3	17	-12	2	-4	-19	20	-7	10	-18	5	-8	1	-6	13	14	-15	-11
t_{10}	-3	6	4	-11	15	12	-18	2	17	-9	-20	8	-16	5	-19	7	1	-13	-14
t_{11}	-1	5	-13	10	19	-7	-16	6	-2	15	-3	17	-14	-12	8	20	-4	-18	9
t_{12}	-5	8	-7	9	18	-10	-3	19	-15	13	-6	14	-1	11	-16	2	17	-20	4
t_{13}	-19	18	11	-1	-7	14	-2	-17	4	-12	15	3	-6	16	20	-9	8	10	-5
t_{14}	-17	1	-15	7	6	-13	-8	3	-20	2	-16	-12	11	-4	5	18	-9	-19	10
t_{15}	-18	20	14	-5	-10	1	-17	-7	12	-11	-13	6	-19	3	-2	-4	16	9	-8
t_{16}	9	-4	-19	20	-5	-18	11	-1	-3	6	14	-7	10	-13	12	-17	-15	8	2
t_{17}	14	-2	-9	4	8	-6	15	13	-10	3	5	-11	20	-18	1	16	-12	-7	19
t_{18}	15	-13	8	-3	-12	16	10	-4	1	-5	9	-20	2	17	-7	-14	19	11	-6
t_{19}	13	-7	16	-6	-11	3	9	-12	5	-8	4	-2	15	-20	10	-1	-18	14	-17
t_{20}	2	-15	5	-16	-4	8	7	-9	14	-1	10	18	-17	19	-13	-11	6	12	-3

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	5	-13	4	8	-3	2	-7	15	-16	-11	10	14	-12	9	-20	18	17	-19	6
t_2	-13	8	-19	-7	9	-1	16	-5	10	20	-3	-17	18	-6	14	-11	-15	12	-4
t_3	-12	-18	13	-11	1	-6	-20	19	14	-10	2	9	-7	15	17	-16	-4	8	-5
t_4	6	17	-1	19	-20	10	12	-9	-18	8	-11	-16	5	-14	-7	13	3	-15	2
t_5	-1	-15	9	17	-16	20	-13	2	-8	-12	7	11	-4	10	-18	19	14	-6	3
t_6	-4	-19	15	-12	14	3	-18	-17	11	-7	20	10	-13	2	16	-8	-9	5	-1
t_7	20	14	-16	2	-13	-12	1	-11	-15	6	-5	-19	3	-8	4	-9	-18	10	-17
t_8	-14	-2	10	-1	17	18	-15	20	5	-4	13	12	-9	7	-19	6	11	-3	16
t_9	19	12	-5	18	-2	-17	11	4	-20	16	-14	-3	8	-1	-10	7	6	-13	15
t_{10}	18	11	-8	20	-15	-4	14	-12	-2	3	-1	-6	16	-5	9	-17	19	-7	13
t_{11}	16	-10	-17	3	-19	13	-9	7	-6	1	4	-5	14	12	-15	2	-8	-20	18
t_{12}	3	-9	-14	6	-18	7	-4	10	-19	5	-17	-8	1	-11	-13	15	16	-2	20
t_{13}	2	1	-3	-15	7	-11	5	-14	17	19	-8	-18	6	-16	12	-4	-20	9	-10
t_{14}	8	-7	12	16	-6	15	-10	13	-3	17	9	-1	-11	4	-2	20	-5	-18	19
t_{15}	17	5	-6	13	10	-14	8	-1	7	18	-16	-20	19	-3	11	-12	2	4	-9
t_{16}	-11	-20	7	-14	5	19	-2	18	1	-9	15	4	-10	13	-6	3	-12	17	-8
t_{17}	-15	-4	11	-5	-8	9	-19	6	-13	-14	12	2	-20	18	-3	10	-1	-16	7
t_{18}	-10	3	20	-9	12	-8	6	-16	4	-15	-19	13	-2	-17	5	-1	7	14	-11
t_{19}	-9	6	2	-4	11	-16	17	-3	12	-13	18	7	-15	20	8	-5	-10	1	-14
t_{20}	-7	16	-18	-10	4	-5	3	-8	9	-2	-6	15	17	-19	1	-14	13	11	-12

2011/12

Table H9: 2011/12 best found schedule

team, slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
t_1	-20	5	18	-19	7	-12	-4	8	2	-15	9	-17	-11	13	16	-3	14	6	-10
t_2	12	-14	16	-10	4	6	-3	20	-1	11	5	-9	18	-8	-13	17	-15	19	-7
t_3	18	-4	-5	20	-6	16	2	-11	-8	17	-7	10	9	-15	-14	1	-19	-13	12
t_4	-16	3	-14	12	-2	-13	1	-18	20	-9	19	-7	-5	11	15	-10	-17	8	6
t_5	14	-1	3	-16	8	15	-20	19	-18	7	-2	6	4	-9	-17	12	10	-11	-13
t_6	-19	15	-13	7	3	-2	17	-12	-10	18	-14	-5	16	20	-8	9	11	-1	-4
t_7	13	-17	15	-6	-1	8	-10	16	12	-5	3	4	-14	-18	11	-19	-9	20	2
t_8	-10	16	-12	17	-5	-7	14	-1	3	-19	-18	11	-20	2	6	-15	13	-4	9
t_9	-15	10	-17	13	20	-11	12	-14	-16	4	-1	2	-3	5	19	-6	7	-18	-8
t_{10}	8	-9	-11	2	-19	-20	7	-13	6	14	-17	-3	15	-16	-18	4	-5	-12	1
t_{11}	17	-12	10	-15	-18	9	-16	3	14	-2	20	-8	1	-4	-7	13	-6	5	19
t_{12}	-2	11	8	-4	17	1	-9	6	-7	-16	15	14	-13	19	20	-5	18	10	-3
t_{13}	-7	19	6	-9	-14	4	-15	10	-17	-20	16	-18	12	-1	2	-11	-8	3	5
t_{14}	-5	2	4	-18	13	19	-8	9	-11	-10	6	-12	7	-17	3	-20	-1	15	-16
t_{15}	9	-6	-7	11	16	-5	13	-17	19	1	-12	20	-10	3	-4	8	2	-14	18
t_{16}	4	-8	-2	5	-15	-3	11	-7	9	12	-13	19	-6	10	-1	18	20	-17	14
t_{17}	-11	7	9	-8	-12	18	-6	15	13	-3	10	1	-19	14	5	-2	4	16	-20
t_{18}	-3	20	-1	14	11	-17	-19	4	5	-6	8	13	-2	7	10	-16	-12	9	-15
t_{19}	6	-13	-20	1	10	-14	18	-5	-15	8	-4	-16	17	-12	-9	7	3	-2	-11
t_{20}	1	-18	19	-3	-9	10	5	-2	-4	13	-11	-15	8	-6	-12	14	-16	-7	17

team, slot	s_{20}	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}	s_{31}	s_{32}	s_{33}	s_{34}	s_{35}	s_{36}	s_{37}	s_{38}
t_1	-9	4	-7	12	-2	19	-18	-13	15	-8	10	-14	-6	3	17	-16	11	20	-5
t_2	-5	3	-4	-6	1	10	-16	8	-11	-20	7	15	-19	-17	9	13	-18	-12	14
t_3	7	-2	6	-16	8	-20	5	15	-17	11	-12	19	13	-1	-10	14	-9	-18	4
t_4	-19	-1	2	13	-20	-12	14	-11	9	18	-6	17	-8	10	7	-15	5	16	-3
t_5	2	20	-8	-15	18	16	-3	9	-7	-19	13	-10	11	-12	-6	17	-4	-14	1
t_6	14	-17	-3	2	10	-7	13	-20	-18	12	4	-11	1	-9	5	8	-16	19	-15
t_7	-3	10	1	-8	-12	6	-15	18	5	-16	-2	9	-20	19	-4	-11	14	-13	17
t_8	18	-14	5	7	-3	-17	12	-2	19	1	-9	-13	4	15	-11	-6	20	10	-16
t_9	1	-12	-20	11	16	-13	17	-5	-4	14	8	-7	18	6	-2	-19	3	15	-10
t_{10}	17	-7	19	20	-6	-2	11	16	-14	13	-1	5	12	-4	3	18	-15	-8	9
t_{11}	-20	16	18	-9	-14	15	-10	4	2	-3	-19	6	-5	-13	8	7	-1	-17	12
t_{12}	-15	9	-17	-1	7	4	-8	-19	16	-6	3	-18	-10	5	-14	-20	13	2	-11
t_{13}	-16	15	14	-4	17	9	-6	1	20	-10	-5	8	-3	11	18	-2	-12	7	-19
t_{14}	-6	8	-13	-19	11	18	-4	17	10	-9	16	1	-15	20	12	-3	-7	5	-2
t_{15}	12	-13	-16	5	-19	-11	7	-3	-1	17	-18	-2	14	-8	-20	4	10	-9	6
t_{16}	13	-11	15	3	-9	-5	2	-10	-12	7	-14	-20	17	-18	-19	1	6	-4	8
t_{17}	-10	6	12	-18	-13	8	-9	-14	3	-15	20	-4	-16	2	-1	-5	19	11	-7
t_{18}	-8	19	-11	17	-5	-14	1	-7	6	-4	15	12	-9	16	-13	-10	2	3	-20
t_{19}	4	-18	-10	14	15	-1	20	12	-8	5	11	-3	2	-7	16	9	-17	-6	13
t_{20}	11	-5	9	-10	4	3	-19	6	-13	2	-17	16	7	-14	15	12	-8	-1	18

Solution Quality

Table H10 provides a breakdown of the accumulated weighted penalties associated with the official (o), proposed (p) and best (b) schedules for the three seasons considered. Note that constraint C103 is violated exactly once for each of the best schedules and on no occasion by the proposed schedules. Costs referred to in the report are in bold font.

Table H10: Schedule total cost breakdown

Constraints			Season								
			2013/14 penalty			2012/13 penalty			2011/12 penalty		
constraint	description	weight	o	p	b	o	p	b	o	p	b
Home and away venue											
C04	Club away request	60	420	360	360	420	420	420	420	360	420
C07	Semi-complementary	100	0	0	0	0	0	0	**200**	**100**	0
Strength and paired team											
C102a	First slot strength	80	0	0	0	80	80	0	0	0	0
C102b	Holiday strength	90	90	90	0	90	90	0	90	90	180
C102c	Strength prior UCL	45	585	450	225	630	225	135	450	180	90
C102d	Strength prior UEL	5	25	25	5	40	25	25	10	20	10
C102e	Strength after UCL	30	540	360	510	570	330	330	510	360	420
C102f	Strength after UEL	9	18	27	36	27	45	81	27	18	27
C103	Holiday paired	100	0	0	**100**	0	0	**100**	100	0	**100**
Breaks											
C12a	First two alt.	90	0	0	0	0	0	0	0	0	0
C12b	Last two alt.	90	0	0	0	0	0	0	0	0	0
C12c (hard)	Holiday break	1000	0	0	0	0	0	0	0	0	0
C13, C14	Pattern break	100	0	0	0	0	0	0	0	0	0
C105	FA break	2	32	40	20	4	22	18	28	20	28
C106	Bank Holiday break	50	300	0	0	200	0	0	200	0	0
Tournament quality											
C104 (hard)	2RR	1000	0	0	0	**20000**	0	0	0	0	0
C107	Min holiday dist.	0.5/km	717	222	157	368	377	25	503	386	144
C108 (hard)	Holiday reverse	1000	0	0	0	0	0	0	0	0	0
C19	Reverse proximity	60	0	0	0	0	0	0	0	0	0
Feasibility											
C110 (hard)	HAP Set SumRow	M	0	0	0	0	0	0	0	0	0
C111 (hard)	HAP Set SumCol	M	0	0	0	0	0	0	0	0	0
C112 (hard)	T Set SumRow	M	0	0	0	0	0	0	0	0	0
C113 (hard)	T Set SumCol	M	0	0	0	0	0	0	0	0	0
	Total violations		68	**57**	44	79	**46**	45	61	**40**	45
	Holiday Distance [km]		4120	**3130**	3000	3066	**3084**	2380	3074	**2840**	2355
	Total Z cost		2727	**1574**	1413	22429	**1614**	1134	2538	**1534**	1419

APPENDIX I

VERIFY AND VALIDATE SPREADSHEET

An electronic copy of an Excel document called ‘Validate and verify.xlsx’ is included on the CD. This document describes all verification processes followed in this research. Tables showing the validation process followed are also shown.