



Scheduling English football fixtures over holiday periods

G Kendall*

The University of Nottingham, Nottingham, UK

Every year the English football authorities produce a set of fixtures for the four main divisions in England. Over the Christmas and New Year period every team has to play two fixtures; one being played at their home venue and the other at an opponent's venue. There are various other constraints that also have to be respected with the overall objective being to minimize the total distance travelled by all teams. In this paper, I formally define the problem, discuss the data collection that I have undertaken and present the algorithm (which is based on depth first search, followed by a local search) I have developed. Using data from four seasons, I show that I am able to produce better schedules than those currently used.

Journal of the Operational Research Society (2008) 59, 743–755. doi:10.1057/palgrave.jors.2602382

Published online 7 March 2007

Keywords: scheduling; sports; optimization

Introduction

'Football supporters and motoring organizations have called for changes in the scheduling of fixtures [over the Christmas holiday period] to prevent supporters from having to drive home from matches, well into the early hours of the morning' (*The Times* (Sport, Football), 30th December 2005, p 68).

Football (called soccer in the USA) is the national game of England. There are four main divisions (see Appendix A for the main terms I use in this paper) which most people would consider as making up the '*English football league*'. These divisions are administered by three bodies, the Football Association (FA), the Premier League and the Football League. The Football Association (<http://www.thefa.com>, last accessed 13 March 2006) is the governing body of the game with a key aim of developing the sport at all levels. The Premier League (<http://www.premierleague.com>, last accessed 13 March 2006), formed in May 1992 in order to promote the top division of the game, has commercial independence from the Football Association and the Football League and is able to negotiate its own broadcast and sponsorship agreements. The Football League (<http://www.football-league.co.uk>, last accessed 13 March 2006) provides the referees for fixtures, produces the fixture programme, administers the transfer system and looks for ways to generate extra revenue for its member clubs.

For the 2005–2006 season, the top four divisions were known as the '*Barclays Premiership*', '*Coca-Cola Championship*',

'*Coca-Cola League One*' and '*Coca-Cola League Two*'. The names of the divisions change in line with sponsorship arrangements. The Barclays Premiership comprises 20 teams, with the other three divisions each having 24 teams. Each division is self-contained, in that they hold their own double round robin tournament. The total number of fixtures played in the Barclays Premiership is 380, with 552 fixtures being played in the other three divisions, giving a total of 2036 fixtures that have to be scheduled in a given season.

To produce the required double round robin tournament for each division, I could use algorithms drawn from the literature, with one of the most common being the polygon construction method (Dinitz *et al*, 2007). However, this method is unlikely to produce a fixture list that is acceptable to all interested parties (which include the football authorities, the football clubs, the supporters, the police, etc) as the algorithm does not cater for all the real-world requirements that make up an acceptable set of fixtures. For example, the *pairing system* (see Problem Definition section) operates across divisions, so that fixtures in one division cannot be scheduled without reference to the other divisions.

Another real-world requirement, not captured by methods such as polygon construction, arises when scheduling fixtures over holiday periods. The period around Christmas is a particularly busy time, with teams playing at least twice. The dates on which the fixtures take place over this period differs from year to year, and there could be more than two fixtures for each team. However, it is always the case that each team will play two related fixtures, with these fixtures having to adhere to the set of constraints discussed below (Problem Definition section and in Appendix B). Given the specific requirements of these two fixtures, it is usual to schedule them first and then generate the rest of the schedule around them. The aim of this

*Correspondence: G Kendall, School of Computer Science and IT, The University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, Notts NG8 1BB, UK.

E-mail: gkx@cs.nott.ac.uk

Table 1 Analysis of published fixtures over the holiday period

<i>Season</i>	<i>Date/fixtures played/distance</i>
2005–2006	December 23/1/52, December 26/45/4243, December 28/46/6336 , December 31/46/4503, January 02/43/6564, January 03/3/296
2004–2005	December 26/46/4563 , December 28/45/6164, December 29/1/285, January 01/46/5122, January 03/46/7139
2003–2004	December 26/46/3872, December 28/45/4218, December 30/1/78
2002–2003	December 26/46/3827 , December 28/41/6747, December 29/5/839, January 01/46/3964

Bold entries are the ones I refer to as *published fixtures* in this study (format is *date/number of fixtures played/distance*).

paper is to investigate if we are able to generate fixtures, for these two days, that adhere to the various constraints, but have travel distances that are less than the distances of the fixtures produced by the football authorities (I refer to the fixture lists produced by the football authorities as the *published fixtures*). Note that I will refer to the first fixture list as the *Boxing Day* (26th December) fixtures and the second fixture list as the *New Year's Day* (1st January) fixtures. As can be seen from Table 1, the fixtures are not always played on these two dates but it is convenient to refer to them in this way.

This paper is organized as follows. In the next section, I present related work with respect to sports scheduling. Next, I informally describe the problem (a formal definition is given in Appendix B). Following this, I present the data that I have used. Next, I describe the algorithmic approach that I adopt. Finally, I show the results and then present my conclusions and suggest future directions for research in this area.

Related work

There is a body of work which has considered the minimization of travel distances for sports schedules. Costa (1995), for example, investigated the scheduling requirements of the National Hockey League, where one of the factors was to minimize the distances travelled. The introduction of the Travelling Tournament Problem (Easton *et al.*, 2001), using distances based on the road trips that have to be undertaken by Major League Baseball teams in the United States, has helped promote research interest in finding shortest travelling distances. Easton *et al.* initially introduced seven instances, containing 4, 6, 8, 10, 12, 14 and 16 teams, which have been tackled, using various approaches. See, for example Crauwels and van Oudheusden (2002), Ribeiro and Urrutia (2004), Easton *et al.* (2003) and Araújo *et al.* (2006). The best results have been reported, using simulated annealing (Anagnostopoulos *et al.*, 2006). A new set of larger instances has recently been introduced (December 2005). This comprises an additional nine instances (16, 18, 20, 22, 24, 26, 28, 30 and 32 teams). An up-to-date list of the best known solutions, as well as details of all the instances, can be found on Michael Trick's web site (<http://mat.gsia.cmu.edu/TOURN/>, last accessed 13 March 2006).

Urrutia and Ribeiro (2004) have shown that minimizing distance and maximizing breaks (two consecutive home

games (home break) or two consecutive away games (away break)) is equivalent. This followed previous work (de Werra, 1981, 1988; Elf *et al.*, 2003) which showed how to construct schedules with the minimum number of breaks.

Much of the previous work (especially with respect to the Travelling Tournament Problem) is based on the assumption that teams go on *road trips*. A road trip involves a team leaving their home city, travelling to their first opponent, playing a fixture and then travelling to their second opponent (and possibly third, fourth, . . .), before returning home. Road trips are common in countries such as the US, due to the distances involved, but not common in the UK where teams will leave their home location, play the fixture and then return home. Therefore, there is a distinct difference between the previous work on the Travelling Tournament Problem, and the work reported here. For a set of fixtures, making up an entire English football season we cannot minimize the total distance travelled. The amount of travel is dictated by the fixtures that are to be played because each team returns home after *every* fixture. However, we can consider other factors of the fixture list in order to assist both the teams and the supporters. For example, we could try and minimize the distance between teams for mid-week fixtures. We might also attempt to minimize the number of breaks and, in an ideal set of fixtures, every team will play alternate home and away fixtures. We could also try and schedule some fixtures when they are likely to receive the most media interest or we could schedule fixtures which are likely to have low supporter interest in order to maximize gate receipts.

The scheduling aspect that I am considering in this paper is minimizing the distance travelled for two complete fixtures (a *complete fixture* is defined as a set of fixtures when every team plays). These two complete fixtures can be used over the Christmas holiday period when, for a variety of reasons, teams wish to limit the amount of travelling they do. As I shall discuss later, these fixtures are normally scheduled first and they do have the lowest travelling distances of all the complete fixtures in the season. In order to judge the effectiveness of my approach, I shall compare my fixtures against the fixtures that were actually used (which I shall refer to as the *published fixtures*).

Readers with an interest in sports scheduling are referred to the following papers which contain problems and approaches applicable to a variety of sports (Campbell and Chen, 1976;

Cain, 1977; Bean and Birge, 1980; Ferland and Fleurent, 1991; Wright, 1991, 1994; Schreuder, 1992; Armstrong and Willis, 1993; Russell and Leung, 1994; Willis and Terrill, 1994; Ball and Webster, 1997; Nemhauser and Trick, 1998; Schaerf, 1999; Thompson, 1999; Henz, 2001; Yang *et al.*, 2002; Burke *et al.*, 2004; Henz *et al.*, 2004; Cassady *et al.*, 2005; Coleman, 2005; Bartsch *et al.*, 2006; Drexler and Knust, 2007).

An excellent overview of sports scheduling can be found in (Easton *et al.*, 2004).

Problem definition

The problem is formally defined in Appendix B. Here, I give an informal description.

The *pairing system* specifies that certain teams should not be allowed to play at home on the same day. Examples of these include *Manchester United* and *Manchester City*, *Liverpool* and *Everton* and *Chelsea* and *Fulham*. These restrictions are typically due to policing requirements. The pairings used in this paper are shown in Appendix C.

I am required to produce two complete fixtures, one for Boxing Day and the other for New Year's Day. Each complete fixture consists of 92 teams, resulting in 46 fixtures. The overall aim is to minimize the total distance travelled by all the teams, over the two days. In addition, there are a number of constraints that have to be respected. These are hard constraints, in that they cannot be violated.

- (1) If a team plays at home on Boxing Day, it must play away on New Year's Day. Similarly, a team playing at home on New Year's Day must play away on Boxing Day.
- (2) The same teams cannot play each other on both Boxing Day and New Year's Day.
- (3) Paired teams cannot play each other. In fact, in the published fixtures, this constraint is actually violated (for example, in the 2005–2006 season Chelsea and Fulham played each other on Boxing Day). However, in this paper, I treat this as a hard constraint and do not allow this constraint to be violated.
- (4) The total number of paired teams playing at home cannot exceed given limits. These limits are set to the same values as those used in published fixtures (see Tables 3 and A1).
- (5) There is a limit on the number of London-based clubs that can play at home on any one day. Similarly, there are limits on the number of London Premiership clubs that can play at home on the same day and also the number of Greater Manchester clubs that can play at home on the same day. The limits of these values are shown in Appendix D.

Data collection and analysis of published fixtures

As mentioned in the introduction, all teams play (at least) two fixtures over the Christmas period. Typically, these fixtures

are played on Boxing Day (26th December) and New Year's Day (1st January), but the actual dates do differ from year to year. Table 1 analyses the published fixtures at this time of the year, for the four seasons which I consider in this paper.

For season 2005–2006, every team played four fixtures over the holiday period. I will compare my results against 23rd December and 26th December as one complete fixture (total distance = $52 + 4243 = 4295$) and 28th December as the other complete fixture. The alternative (31st December, 2nd January and 3rd January) has a greater travel distance than 23rd/26th/28th December which is why I am using the other fixtures. In fact, in 2005/2006, every team has a four fixture sequence of home/away/home/away or away/home/away/home over the holiday period and I discuss this particular season further in my concluding remarks.

In the 2004–2005 season there are fixtures on Boxing Day and New Year's Day, so I have used these for comparison purposes. The other complete fixtures (28th/29th December and 3rd January) also adhere to the constraint that if a team played at home on one day it must play away on the other day (and vice versa). Like the 2005–2006 season, every team has a four fixture sequence of home/away/home/away or away/home/away/home. However, the 28th/29th December and 3rd January fixtures have a higher travel distance ($6164 + 285 = 6449$ and 7139) and so are not used for comparison purposes here.

The 2003–2004 season only has one choice with the New Year's Day fixtures being played on 28th December and 30th December.

In the 2002–2003 season there are fixtures on Boxing Day and New Year's Day, so I have used these for comparison purposes. The other complete fixture (28th December and 29th December) adheres to the constraint that if a team played at home on Boxing Day, it plays away on 28th/29th December (and vice versa). Therefore, every team has a three fixture sequence of home/away/home or away/home/away. However, the 28th/29th December fixtures has a higher travel distance ($6747 + 839 = 7586$) and so is not used for comparison purposes here.

Therefore, in this paper, I have access to four sets of data (seasons 2002–2003, 2003–2004, 2004–2005 and 2005–2006) and Table 2 presents just one of the published fixtures (Boxing Day, 2004–2005 season). The *Fixture* and *Division* columns are self-explanatory. The *Distance* column is the distance between each team and is calculated using the postcode (ZIP code) of the teams and utilizing one of the UK car breakdown service web sites which incorporates a route planner (<http://www.greenflag.co.uk>, accessed 28 December 2005). The distances for this study were all collected at the same time (during December 2005). The distances will actually alter over time due to changes in road layouts, the construction of new roads, etc. With this in mind, I have supplied all the distances used in this paper on the author's web site (due to space limitations it is too large to list here)

Table 2 Published fixtures for Boxing Day fixtures for season 2004–2005 (distances are in miles)

<i>Fixtures</i>	<i>Division</i>	<i>Distance</i>	<i>Clash?</i>
Arsenal <i>versus</i> Fulham	0	11	
Birmingham City <i>versus</i> Middlesbrough	0	172	Coventry
Blackburn Rovers <i>versus</i> Newcastle United	0	128	
Chelsea <i>versus</i> Aston Villa	0	119	
Crystal Palace <i>versus</i> Portsmouth	0	72	
Everton <i>versus</i> Manchester City	0	37	Tranmere
Manchester United <i>versus</i> Bolton Wanderers	0	18	
Norwich City <i>versus</i> Tottenham Hotspur	0	110	Colchester
Southampton <i>versus</i> Charlton Athletic	0	89	
West Bromwich Albion <i>versus</i> Liverpool	0	94	
Brighton & Hove Albion <i>versus</i> Gillingham	1	72	
Cardiff City <i>versus</i> Wolverhampton Wanderers	1	115	
Coventry City <i>versus</i> Sheffield United	1	81	Birmingham
Crewe Alexandra <i>versus</i> Burnley	1	68	
Leicester City <i>versus</i> Rotherham United	1	69	
Millwall <i>versus</i> Ipswich Town	1	83	
Plymouth Argyle <i>versus</i> Queens Park Rangers	1	214	
Reading <i>versus</i> Watford	1	44	
Stoke City <i>versus</i> Preston North End	1	68	
Sunderland <i>versus</i> Leeds United	1	99	
West Ham United <i>versus</i> Nottingham Forest	1	136	
Wigan Athletic <i>versus</i> Derby County	1	86	
Blackpool <i>versus</i> Hull City	2	138	
Bradford City <i>versus</i> Wrexham	2	90	Huddersfield
Brentford <i>versus</i> Torquay United	2	187	
Chesterfield <i>versus</i> Luton Town	2	117	
Colchester United <i>versus</i> Bournemouth	2	180	Norwich
Doncaster Rovers <i>versus</i> Milton Keynes Dons	2	124	
Hartlepool United <i>versus</i> Oldham Athletic	2	116	
Huddersfield Town <i>versus</i> Port Vale	2	76	Bradford
Peterborough United <i>versus</i> Swindon Town	2	112	
Sheffield Wednesday <i>versus</i> Walsall	2	84	
Stockport County <i>versus</i> Bristol City	2	170	
Tranmere Rovers <i>versus</i> Barnsley	2	82	Everton
Boston United <i>versus</i> Lincoln City	3	34	
Bristol Rovers <i>versus</i> Leyton Orient	3	127	
Bury <i>versus</i> Shrewsbury Town	3	77	
Chester City <i>versus</i> Scunthorpe United	3	131	
Darlington <i>versus</i> Rochdale	3	103	
Grimsby Town <i>versus</i> Macclesfield Town	3	107	
Kidderminster Harriers <i>versus</i> Swansea City	3	124	
Mansfield Town <i>versus</i> Cambridge United	3	104	Notts County
Notts County <i>versus</i> Northampton Town	3	64	Mansfield
Rushden & Diamonds <i>versus</i> Oxford United	3	64	
Wycombe Wanderers <i>versus</i> Southend United	3	83	
Yeovil Town <i>versus</i> Cheltenham Town	3	84	

(<http://www.cs.nott.ac.uk/~gxxk/research-data/>). The *Clash* column indicates where there is a pair clash in the published fixtures. That is, two teams are playing at home, when they have been defined as a pair. For example, in Table 2, both Birmingham and Coventry are at home and these teams have been defined as a pair.

Table 2 represents just a sample of the published fixtures. The other fixtures can be found in Rollin and Rollin (2002, 2003, 2004, 2005) but the important factors, with regards to the work presented here, have been summarized in Table 3.

Depth first search and local search algorithm

There are 92 teams across the four divisions, so a complete fixture comprises 46 fixtures. In order to create fixtures for Boxing Day and New Year's Day, I need to create two complete fixtures while minimizing travel distances, limiting the number of pair clashes and adhering to the other constraints described above and formally presented in Appendix B.

One possible way to generate the schedules is to create a complete search tree, where each node represents a fixture and the path from the root node to the bottom of the tree

Table 3 Analysis of fixtures for the past four seasons showing the total distance travelled and the number of pair clashes for Boxing Day and New Year's Day

		<i>Published fixtures</i>	<i>Generated fixtures</i>
<i>Season 2005–2006</i>			
	Total distance	4295	3049
Boxing Day*	Clashes	12	10
New Year's Day [†]	Total distance	6336	3868
	Clashes	14	10
<i>Season 2004–2005</i>			
	Total distance	4563	3725
Boxing Day	Clashes	10	10
New Year's Day	Total distance	5122	3059
	Clashes	10	10
<i>Season 2003–2004</i>			
	Total distance	3872	2859
Boxing Day	Clashes	8	8
New Year's Day [‡]	Total distance	4296	3500
	Clashes	14	14
<i>Season 2002–2003</i>			
	Total distance	3827	2839
Boxing Day	Clashes	10	10
New Year's Day	Total distance	3964	3201
	Clashes	8	8

*One of these fixtures took place on 23rd December.

[†]These fixtures took place on 28th December.

[‡]These fixtures took place on 28th December and 30th December.

represents a complete fixture. In this case, the depth of the search tree will be 45 (assuming the root node is level 0) and the branching factor is as follows. If I start with the Barclays Premiership (20 teams), I would firstly choose two teams (I use the notation $C(m, n)$ to represent ' m choose n '). Therefore, I have $C(20, 2) \times 2 = 190 \times 2 = 380$ nodes at level 0. Note that I multiply by 2, as each fixture can be reversed to switch the home and away teams. The next level would have a branching factor of $C(18, 2) \times 2 = 153 \times 2 = 306$ and so on until I reach $C(2, 2) \times 2 = 1 \times 2 = 2$. This would represent a search tree with 10 levels and $380 \times 306 \times \dots \times 2$ nodes (2.43×10^{18}).

Once I have produced the search tree for the Barclays Premiership, I would now have to continue building the tree by incorporating the Coca-Cola Championship. Therefore, going from level 9 to level 10 would have a branching factor of $C(24, 2) \times 2 = 276 \times 2 = 552$, with the next level having a branching factor of $C(22, 2) \times 2 = 231 \times 2 = 462$ nodes. Once the tree had been expanded for the Coca-Cola Championship, I would repeat the process for the two remaining divisions. The size of the search tree for just one of the Coca-Cola divisions is of the order 6.20×10^{23} .

A full enumeration of a search tree for just a single division is not possible, let alone a combined search tree for all the divisions. Therefore, I have developed a two-stage approach. The first stage builds a *separate* search tree for each division, incorporating pruning to reduce the size of the search space. Each division returns the best solution it finds which is a set of 10 fixtures (resp. 12) for the Barclays Premiership

(respectively the Coca-Cola divisions). This search process is carried out twice for each division, once for Boxing Day and once for New year's Day. There are slight differences between the two searches for the different days, which I describe below.

The second stage carries out a local search across all divisions, for both days, in order to find a good quality, feasible solution. The details of both these stages are described below.

Stage 1: Depth First Search: I initially produce a solution for *each* division, for both days. I generate a search tree, pruning at branches that either cannot lead to a better solution than one already generated or which will lead to an infeasible solution.

I begin by taking one team (at random) and generating all its possible fixtures (both home and away). Next I carry out a depth first search. A node is expanded by generating all possible fixtures. In expanding a node, I check the following.

- (1) I do not expand a node that would lead to the same team playing twice in that branch of the tree, as this would be an infeasible schedule.
- (2) If the travel distance for two teams is greater than 200 miles, I do not generate that search node. The figure of 200 is based on the fact that (in the UK) this represents travel time of about 3 h which is about the maximum that teams/supporters would wish to travel over the holiday period. I do make an exception for *Plymouth*. Owing to its location there are not many teams within 200 miles so I raise their limit to 250.

- (3) If the expanded node would lead to a *total* travel distance which is greater than the best solution seen so far, I do not generate that node. The best solution found so far is defined as a node which contains a full set of fixtures (ie at level 9 for the Barclays Premiership and level 11 for the other divisions) and the total distance travelled is the minimum found so far.
- (4) I do not allow paired teams to play each other.
- (5) I do not allow paired teams to both be playing at home.

I follow exactly the same procedure as above to generate fixtures for New Year's Day. In addition, the complete fixture found above is input to this search procedure so that additional checks can be performed. In this procedure, I carry out these additional checks when generating new nodes in this search tree.

- (1) I relax the condition (5, above) that paired teams cannot both be at home. I found that if I did not do this, I did not generate any feasible solutions for New Year's Day.
- (2) I do not allow the teams to play each other again. That is, if two teams played each other on Boxing Day, they cannot play each other again on New Year's Day.
- (3) If a team plays at home (respectively away) on Boxing Day, they cannot play at home (respectively away) on New Year's Day.

Using the above procedure I generate two complete fixtures (one for Boxing Day and one for New Year's Day). These fixtures are significant improvements over the published fixtures with regards to the minimization of the total travel distance but I have not yet taken into account any of the pairing constraints, or London/Manchester constraints, as these constraints operate across divisions. The local search stage addresses this.

Stage 2: Local Search: The output of stage 1 are two complete fixtures, $f(B)$ (Boxing Day) and $f(N)$ (New Year's Day). Both $f(B)$ and $f(N)$ contain 46 fixtures and there are no fixtures where the same teams play each other twice. Also, a team playing at home in $f(B)$, will be playing away in $f(N)$, and vice versa.

The aim of the local search is to reduce the number of pair clashes to acceptable values (ie less than or equal to those of the published lists) as well as to ensure that the other constraints are not violated. Before describing the local search procedure I define a swap operation which I use within the search.

Swap home and away

This operation swaps the home and away teams in a given schedule. If I were only considering one complete fixture, this would be a simple operation as simply swapping the teams in a single fixture would still lead to a feasible schedule and, in addition, it would not affect the distance travelled as the distances are assumed to be symmetric. However, where I

a			
...
Chelsea	Portsmouth	Arsenal	Chelsea
...
...	...	Portsmouth	Fulham
...
Fixture $f(B)$		Fixture $f(N)$	

b			
...
...	...	Arsenal	Portsmouth
Portsmouth	Chelsea
...
...	...	Chelsea	Fulham
...
Fixture $f(B)$		Fixture $f(N)$	

Figure 1 (a) The state of two fixtures before the *Swap Home and Away* operation. (b) The state of two fixtures after the *Swap Home and Away* operation.

have two complete fixtures, as I do here, swapping the home and away team in one schedule leads to an infeasible schedule as I now have the same teams which are playing at home (or away) in both complete fixtures. Therefore, I have to carry out the swap in one fixture and repair the other fixture. An example is shown in Figure 1.

Figure 1a shows the state of two schedules ($f(B)$ and $f(N)$) before the swap operation. The fixture *Chelsea* versus *Portsmouth*, in $f(B)$, has been identified as the fixture where the teams will be swapped. After the swap (Figure 1b) the two teams have been swapped. However, in order to maintain feasibility of the two fixtures, I have to locate *Chelsea* and *Portsmouth* in $f(N)$ and swap them as well. Note that this second swap has the potential to alter the travel distances as *Chelsea* were travelling to *Arsenal*, but *Portsmouth* are now travelling to *Arsenal*. *Fulham* were travelling to *Portsmouth* but now *Fulham* are travelling to *Chelsea*.

Having defined this swap operation, I can now describe the local search algorithm. I call the following operators 5000 times (ie a total of 40000 calls). I use the value of 5000 after experimentation but it can easily be increased as the local search moves are computationally inexpensive (the local search takes little more than a few minutes).

- (1) If there is a pair clash in $f(B)$, I locate the fixture that has a maximum number of clashes, choosing randomly if there is more than one candidate. I swap the home and away teams, making the corresponding swap in $f(N)$ in order to maintain feasibility.
- (2) Same as 1 but I now consider $f(N)$.
- (3) If the number of London teams playing at home in $f(B)$ exceeds the threshold, I choose (at random) one of those fixtures and swap the home and away teams. I make the corresponding swap in $f(N)$ to maintain feasibility.
- (4) If the number of Greater Manchester teams playing at home in $f(B)$ exceeds the threshold, I choose (at random) one of those fixtures and swap the home and away teams. I make the corresponding swap in $f(N)$ to maintain feasibility.

Table 4 Summary of improvements over published fixtures

<i>Season</i>	<i>Total distance for published fixtures</i>	<i>Total distance for generated fixtures</i>	<i>Distance saving</i>	<i>Number of pair clashes for published fixtures</i>	<i>Number of pair clashes for generated fixtures</i>
2005–2006	10631	6917	3714 (34.94%)	26	20
2004–2005	9685	6784	2901 (29.95%)	20	20
2003–2004	8168	6359	1809 (22.15%)	22	22
2002–2003	7791	6040	1751 (22.47%)	18	18

The total distance figures are calculated by adding together the Boxing Day and New Year's Day Travel Distances. The distance saving percentage improvement is calculated as '100—generated distance * 100/published distance'.

- (5) If the number of Barclays Premiership London clubs playing at home in $f(B)$ exceeds the threshold, I choose (at random) one of those fixtures and swap the home and away teams. I make the corresponding swap in $f(N)$ to maintain feasibility.
- (6) Same as 3 but now considering $f(N)$.
- (7) Same as 4 but now considering $f(N)$.
- (8) Same as 5 but now considering $f(N)$.

After each of the above, I check if I have a new best solution (ie minimum travel distance and no constraint violations).

At the end of the local search phase, I carry out a post-processing phase, to ensure that a feasible solution is returned. This post-search process checks that I do not have any reverse fixtures (team i playing team j , followed by team j playing team i) and that I do not have any local derbies (two paired teams playing each other). While these two conditions exist, I carry out the swap operator, ensuring I include one of the teams that is contributing to the infeasible solution. The output of this post-process phase is a solution that does not violate any constraints.

Results

Using the above two-stage approach, I am able to produce significant improvements over the published fixtures (see Tables 3 and 4).

Table 3 shows the improvements I achieved for each day. In addition, I was able to reach these solutions with the same, or fewer, pair clashes when compared with the published fixtures. On Boxing Day, for season 2005–2006, for example, I managed a distance saving of 1246 (4295 – 3049), with two less pair clashes. The previous season (2004–2005), the distance saving for Boxing Day is 838 (4563 – 3725) but now achieved with the same number of pair clashes.

Table 4 shows total distance travelled over the two complete fixtures for each season. It is noticeable that the highest savings were achieved in the 2005–2006 season. I discuss the possible reasons for this in my concluding remarks. However, even removing this year from consideration, I achieve an average overall saving of 24.86% ($(29.95 + 22.15 + 22.47)/3$), with the same number of pair clashes.

Table 5 shows one of the complete fixtures that were generated.

Conclusions and discussion

In this paper, I have described a sports scheduling problem that has not been reported before in the literature. The problem is to minimize the total distance travelled by 92 football league clubs over two complete fixtures. At first sight, it may seem to be a similar problem to the Travelling Tournament Problem but, in fact, the problems are very different. The aim of the Travelling Tournament Problem is to minimize distances by finding a set of suitable road trips, so that when the teams play a number of away games, they limit the amount of travel that is undertaken. In this problem, the travel distances, for a given team over a given season, is fixed. Instead, I try and minimize the distance for two dates so that these fixtures can be used over the Christmas period.

As can be seen from the results, I have made significant savings over the published fixtures. In doing so, I have not increased the number of pair clashes and, in one of the seasons, I have actually reduced the number of pair clashes. In addition, I have not allowed any paired teams to play each other. This is occasionally allowed in the published fixtures, which can help the reduction of the overall distance as paired teams are often close to one another. Finally, I have also ensured that I do not have more than a certain number of teams playing in London or Greater Manchester on any one day.

I am currently discussing my results with the Football League. I would like to extend this work in order to try and produce fixture lists for the entire season. This will include catering for many more constraints, which is why I would like to discuss the problem with the problem owners as I would like to ensure I am solving the problem that they actually face and not some simplified abstraction.

With regards to the work reported in this paper, I see a number of future directions that I could follow. The first is to provide a fair comparison for seasons where four complete fixtures are needed over the holiday season. The 2005–2006 season is an example when this was a requirement. Table 1 shows that over the holiday period for this season, there were actually four complete fixtures played. The two additional complete fixtures (played on 31st December, 2nd

Table 5 Generated fixtures for Boxing Day fixtures for season 2004–2005 (distances are in miles)

<i>Fixtures</i>	<i>Division</i>	<i>Distance</i>	<i>Clash?</i>
Aston Villa <i>versus</i> Fulham	0	126	Coventry
Chelsea <i>versus</i> Portsmouth	0	71	
Crystal Palace <i>versus</i> Birmingham City	0	127	
Everton <i>versus</i> Manchester City	0	37	Liverpool
Liverpool <i>versus</i> Blackburn Rovers	0	39	Everton
Manchester United <i>versus</i> Middlesbrough	0	122	
Newcastle United <i>versus</i> Bolton Wanderers	0	153	
Norwich City <i>versus</i> Arsenal	0	114	Southend
Tottenham Hotspur <i>versus</i> Charlton Athletic	0	18	
West Bromwich Albion <i>versus</i> Southampton	0	147	
Burnley <i>versus</i> Sunderland	1	112	
Cardiff City <i>versus</i> Brighton & Hove Albion	1	189	
Coventry City <i>versus</i> Wolverhampton Wanderers	1	37	Aston Villa
Derby County <i>versus</i> Nottingham Forest	1	17	
Leeds United <i>versus</i> Wigan Athletic	1	60	Huddersfield
Leicester City <i>versus</i> Sheffield United	1	66	
Millwall <i>versus</i> Queens Park Rangers	1	11	
Reading <i>versus</i> Plymouth Argyle	1	199	
Rotherham United <i>versus</i> Preston North End	1	78	
Stoke City <i>versus</i> Crewe Alexandra	1	17	
Watford <i>versus</i> Ipswich Town	1	91	Luton
West Ham United <i>versus</i> Gillingham	1	33	
Blackpool <i>versus</i> Port Vale	2	82	
Brentford <i>versus</i> Colchester United	2	77	
Bristol City <i>versus</i> Bournemouth	2	81	
Hartlepool United <i>versus</i> Tranmere Rovers	2	157	
Huddersfield Town <i>versus</i> Chesterfield	2	53	Leeds
Hull City <i>versus</i> Bradford City	2	69	
Luton Town <i>versus</i> Milton Keynes Dons	2	22	Watford
Oldham Athletic <i>versus</i> Barnsley	2	42	
Peterborough United <i>versus</i> Doncaster Rovers	2	89	
Stockport County <i>versus</i> Sheffield Wednesday	2	39	
Swindon Town <i>versus</i> Torquay United	2	138	
Wrexham <i>versus</i> Walsall	2	71	
Boston United <i>versus</i> Lincoln City	3	34	
Bury <i>versus</i> Kidderminster Harriers	3	110	
Cheltenham Town <i>versus</i> Swansea City	3	106	
Darlington <i>versus</i> Mansfield Town	3	113	
Macclesfield Town <i>versus</i> Rushden & Diamonds	3	119	
Northampton Town <i>versus</i> Cambridge United	3	61	
Notts County <i>versus</i> Chester City	3	92	
Scunthorpe United <i>versus</i> Grimsby Town	3	33	
Shrewsbury Town <i>versus</i> Rochdale	3	83	
Southend United <i>versus</i> Oxford United	3	105	Norwich
Wycombe Wanderers <i>versus</i> Leyton Orient	3	42	
Yeovil Town <i>versus</i> Bristol Rovers	3	43	

January and 3rd January) also adhere to the constraint that if you play at home on one of these days, then you have to play away on the other day. Also, by looking at the distances ($4503 + 6564 + 296 = 11\,363$) I note that these are also among the lowest in the season when considering all the complete fixtures. This indicates that, for this season, the schedulers have had to produce two sets of complementary fixtures for the holiday period. In fact, this has been reported in the press and is because 2006 is a world cup year and the English football authorities try to schedule games in such a way that allows the English national team to spend more time together

in preparation for the world cup. In the previous season (2004–2005) there were also four fixtures scheduled, which also adhered to the home/away constraints. However, the distances for the alternative fixtures are high so I assume that there was not a requirement to minimize distances. Nevertheless, it will still be interesting to see if I can create a fixture list for this season incorporating four complete fixtures which minimizes distances; if nothing else to validate the method when applied to the 2005–2006 season. My future work will, therefore, investigate if I can produce four complete fixtures that are suitable for use over the holiday period.

I would also like to further investigate the bounds that I place on my depth first search. In the algorithm section, I described how I control whether I expand a node or not. I have some more ideas as to how I can improve this. One idea is to increase the 200 mile limit (250 for Plymouth) so that I consider more solutions. Then, in the local search I could introduce an operator that looks at the team which has the highest travel distance and carry out a move which tries to re-schedule that team. Another idea is to order the fixtures when considering the expansions. That is, to expand lower cost (those with lower distance) nodes first. This might have the effect of reducing the search space as I am likely to exceed the total travel distance (ie best solution found so far) higher in the search tree as we should find a relatively low cost solution early in the search which will reduce the number of expansions I have to carry out later.

It would also be interesting to investigate if I can reduce the number of clashes, while still maintaining the travel distances to acceptable levels. In this current study, I set the upper bound on the number of pair clashes to those that are present in the published fixtures. What effect would it have on the solutions if I decreased these limits? Would the solutions be considered better by, say, the police as, although, the teams would have to travel further, would there be a reduction in the required policing resources due to having fewer teams playing in a given geographical area? In order to investigate this particular line of research, I would need to talk to police authorities, as well as the football authorities.

This has been a very interesting problem to work on and it has the potential for significant further investigation. This study has used an exact method (depth first search) combined with a local search. If I am to tackle the problem of generating fixture lists for the complete season I am likely to have to resort to meta-heuristic approaches (eg genetic algorithms (Goldberg, 1989; Sastry *et al*, 2005), simulated annealing (Kirkpatrick *et al*, 1983; Aarts *et al*, 2005), hyper-heuristics (Burke *et al*, 2003a,b; Ross, 2005; Burke *et al*, 2007) etc). However, as I have shown here, it is possible to decompose the problem and, perhaps, the hybridization of exact and meta-heuristic approaches might also be applicable for the larger problem.

Statement of contribution

This paper, for the first time in the academic literature, studies the problems associated with scheduling football fixtures in the four major English divisions. In particular, I study how fixtures can be scheduled over holidays periods when travel distances have to be minimized.

One contribution of this paper is to generate fixtures which are better than *published fixtures* (ie those that are published and used). By ‘better’, I mean when comparing the overall distance travelled, while still adhering to the required constraints. I use a combination of depth first search, followed by a local search in order to find feasible solutions.

A further contribution of this work describes the problem both in a descriptive way and also presents a formal model. It is hoped that this will promote more research on this problem. In addition, I describe the data collection that was undertaken in order to allow other researchers access to the data in order to reproduce my results or to develop this work even further. Indeed, the data I present could be used for a variety of other applications such as vehicle routing or travelling salesman.

Acknowledgements—I would like to thank the anonymous reviewers for their comments. This has led to a much improved paper.

References

- Aarts E, Korst J and Michiels W (2005). Simulated annealing. In: Burke EK and Kendall G (eds). Chapter 7 of *Introductory Tutorials in Optimization, Decision Support and Search Methodology*. Springer-Verlag: New York, USA, pp 187–210.
- Anagnostopoulos A, Michel L, van Hentenryck P and Vergados Y (2006). A simulated annealing approach to the traveling tournament problem. *J Schedul* **9**: 177–193.
- Araújo APF, Boeres C, Rebello VEF, Ribeiro CC and Urrutia S (2006). Exploring grid implementations of parallel cooperative metaheuristics: A case study for the mirrored traveling tournament problem. In: *Proceedings of the 6th Metaheuristics International Conference*, Vienna, Austria, 2005.
- Armstrong J and Willis RJ (1993). Scheduling in the Cricket World Cup: A case study. *J Opl Res Soc* **44**: 1067–1072.
- Ball BC and Webster DB (1997). Optimal schedules for even-numbered team athletic conferences. *AIIE Trans* **9**: 161–169.
- Bartsch T, Drexl A and Kröger S (2006). Scheduling the professional soccer leagues of Austria and Germany. *Comput Opns Res* **33**: 1907–1937.
- Bean JC and Birge JR (1980). Reducing travelling costs and player fatigue in the National Basketball Association. *Interfaces* **10**: 98–102.
- Burke EK, de Werra D, Landa Silva JD and Raess C (2004). Applying heuristics methods to schedule sports competitions on multiple venues (Extended Abstract). In: *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT '04)*, Pittsburgh, USA, pp 451–456, ISBN 0-88748-413-1.
- Burke E, Hart E, Kendall G, Newall J, Ross P and Schulenburg S (2003a). Hyper-heuristics: An emerging direction in modern search technology. Glover FW and Kochenberger GA (eds). *Handbook of Meta-Heuristics*. Kluwer: Boston, USA, pp 457–474.
- Burke EK, Kendall G and Soubeiga E (2003b). A tabu-search hyper-heuristic for timetabling and rostering. *J Heurist* **9**: 451–470.
- Burke EK, McCollum B, Meisels A, Petrovic S and Qu R (2007). A graph-based hyper-heuristic for educational timetabling problems. *Eur J Opl Res* **176**: 177–192.
- Cain Jr. WO (1977). A computer-assisted heuristic approach used to schedule the major league baseball clubs. Ladany SP and Machol RE (eds). *Optimal Strategies in Sports*. North-Holland Publishing Company: New York, USA, pp 32–41.
- Campbell RT and Chen DS (1976). A minimum distance basketball scheduling problem. Ladany SP, Machol RE and Morrison DG (eds). *Management Science in Sports*. North-Holland Publishing Company: New York, USA, pp 15–26.
- Cassady CR, Maillart LM and Salman S (2005). Ranking sports teams: A customizable quadratic assignment approach. *Interfaces* **35**: 497–510.

- Coleman BJ (2005). Minimizing game score violations in college football rankings. *Interfaces* **35**: 483–496.
- Costa D (1995). An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR: Inform Syst Opl Res* **33**: 161–178.
- Crauwels H and van Oudheusden D (2002). A generate-and-test heuristic inspired by ant colony optimization for the traveling tournament problem. In: *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT '02)*, Gent, Belgium, pp 314–315, ISBN 90-806096-1-7.
- de Werra D (1981). Scheduling in sports. Hansen P (ed). *Studies on Graphs and Discrete Programming*. Elsevier: Amsterdam, pp 381–395.
- de Werra D (1988). Some models of graphs for scheduling sports competitions. *Disc Appl Math* **21**: 47–65.
- Di Gaspero L and Schaerf A (2005). A tabu search approach to the traveling tournament problem. In: *Proceedings of the 6th Metaheuristics International Conference*, Vienna, Austria, 2005.
- Dinitz JH, Froncek D, Lamken ER and Wallis WD (2007). Scheduling a tournament. Colbourn CJ and Dinitz JH (eds). *The CRC Handbook of Combinatorial Designs*. CRC Press Inc.: Florida, USA, (previous edition of this book was published in 1996), pp 591–606.
- Drexel A and Knust S (2007). Sports league scheduling: Graph- and resource-based models. *Omega* **35**: 465–471.
- Easton K, Nemhauser G and Trick M (2001). The traveling tournament problem: description and benchmarks. In: Walsh T (ed). *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, **2239**. Springer-Verlag: Berlin, Germany, pp 580–584.
- Easton K, Nemhauser G and Trick M (2003). Solving the traveling tournament problem: A combined integer programming and constraint programming approach. In: Burke E and DeCausmaecker P (eds). *Selected Revised Papers from the 4th International Conference on the Practice and Theory of Automated Timetabling*, Lecture Notes in Computer Science, **2740**. Springer-Verlag: Berlin, Germany, pp 100–109.
- Easton K, Nemhauser G and Trick M (2004). Sports scheduling. In: Leung JT (ed). *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, Chapter 52, CRC Press Inc: Florida, USA, pp 1–19.
- Elf M, Jünger M and Rinaldi G (2003). Minimizing breaks by maximizing cuts. *Opns Res Lett* **31**: 343–349.
- Ferland JA and Fleurent C (1991). Computer aided scheduling for a sports league. *INFOR: Inform Syst Opl Res* **29**: 14–25.
- Goldberg DE (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley: Boston, MA, USA.
- Henz M (2001). Scheduling a major college basketball conference—revisited. *Opns Res* **49**: 163–168.
- Henz M, Müller T and Thiel S (2004). Global constraints for round robin tournament scheduling. *Eur J Opl Res* **153**: 92–101.
- Kirkpatrick S, Gelatt CD and Vecchi MP (1983). Optimization by simulated annealing. *Science* **220**: 671–680.
- Nemhauser GL and Trick MA (1998). Scheduling a major college basketball conference. *Opns Res* **46**: 1–8.
- Ribeiro CC and Urrutia S (2004). Heuristics for the mirrored traveling tournament problem. In: *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT '04)*, Pittsburgh, USA, pp 323–342, ISBN 0-88748-413-1.
- Rollin G and Rollin J (eds). (2002). *Rothmans Football Yearbook 2002–2003*. Headline Book Publishing: London, UK.
- Rollin G and Rollin J (eds). (2003). *Sky Sports Football Yearbook 2003–2004*. Headline Book Publishing: London, UK.
- Rollin G and Rollin J (eds). (2004). *Sky Sports Football Yearbook 2004–2005*. Headline Book Publishing: London, UK.
- Rollin G and Rollin J (eds). (2005). *Sky Sports Football Yearbook 2005–2006*. Headline Book Publishing: London, UK.
- Ross P (2005). Hyper-heuristics. In: Burke EK and Kendall G (eds). Chapter 17 of *Introductory Tutorials in Optimization, Decision Support and Search Methodology*, Springer-Verlag: New York, USA, pp 529–556.
- Russell RA and Leung JM (1994). Devising a cost-effective schedule for a baseball league. *Opns Res* **42**: 614–625.
- Sastry K, Goldberg D and Kendall G (2005). Genetic algorithms. In: Burke EK and Kendall G (eds). Chapter 4 of *Introductory Tutorials in Optimization, Decision Support and Search Methodology*. Springer-Verlag: New York, USA, pp 97–125.
- Schaerf A (1999). Scheduling sport tournaments using constraint logic programming. *Constraints* **4**: 43–65.
- Schreuder JAM (1992). Combinatorial aspects of construction of competition Dutch professional football leagues. *Disc Appl Math* **35**: 301–312.
- Thompson JM (1999). Kicking timetabling problems into touch. *OR Insight* **12**: 7–15.
- Urrutia S and Ribeiro CC (2004). Minimizing travels by maximizing breaks in round robin tournament schedules. *Electron Notes Disc Math* **18-C**: 227–233.
- Willis RJ and Terrill BJ (1994). Scheduling the Australian state cricket season using simulated annealing. *J Opl Res Soc* **45**: 276–280.
- Wright MB (1991). Scheduling English cricket umpires. *J Opl Res Soc* **42**: 447–452.
- Wright M (1994). Timetabling county cricket fixtures using a form of tabu search. *J Opl Res Soc* **45**: 758–770.
- Yang JT, Huang HD and Horng JT (2002). Devising a cost effective basketball scheduling by evolutionary algorithms. *Proceedings of the 2002 Congress on Evolutionary Computation*. IEEE Press: Piscataway, NJ, USA, pp 1660–1665.

Appendix A. Glossary

Below, I present a glossary of some of the main terms used in this paper, and in the wider context of football. This is to ensure that there is no confusion about how the terms are being used.

Away fixture	Each team will visit every other team's ground over the course of a season. For the travelling team, this is an away fixture.
Division	A set of teams, where every team in the division has to play every other team twice (home and away) over the course of a season. In this paper, I am considering four divisions, one with 20 teams and the other three with 24 teams.
Double round robin tournament	A double round robin tournament is defined as a set of teams (in this case a division) where each team has to play every other team twice, once at home and once away.
Fixture	The meeting of two teams to play a football match.
Home fixture	Each team has a home venue and, throughout the course of a season, they will host every other team in their division at their home ground. For the team playing at home, this is a home fixture.

League	A league is often used interchangeably with <i>division</i> . In this paper, however, I define the term to be a collection of divisions. So, the four divisions considered in this paper make up a league.
Match	This term can be used interchangeably with fixture, although, in this paper, I avoid this term.
Season	The English football season runs from August to May. During this time, teams will often play over the weekend (which could be Saturday, Sunday or Monday; depending on TV deals) and during the week (often Wednesday evening).

Table A1 Limits for the number of clashes allowed for each of the two days in each season

Season	C _{by}	C _{y_y}
2005–2006	12	14
2004–2005	10	10
2003–2004	8	14
2002–2003	10	8

Appendix B. Model

Indices

i 1 . . . *n*
j 1 . . . *n*

where *n* = 92 (the total number of fixtures on Boxing Day and New Year’s Day)

Decision variables

B_i 1 if team *i* is playing at home on Boxing Day, 0 otherwise
Y_i 1 if team *i* is playing at home on New Year’s Day, 0 otherwise
x_{i,j} 1 if team *i* is playing team *j*, 0 otherwise

Parameters

P_{i,j} 1 if team *i* is paired with team *j*, 0 otherwise
D_{i,j} the distance (in miles) between team *i* and team *j*
L_i 1 if team *i* is a London-based club
M_i 1 if team *i* is a Greater Manchester-based club
Q_i 1 if team *i* is a Premier club
Cb_y The number of clashes in the published fixtures for Boxing Day for season, *y*
Cy_y The number of clashes in the published fixtures for New Year’s Days for season, *y*

The values *Cb_y* and *Cy_y* are defined as follows Table A1.

β_l The maximum number of clubs based in London which can play at home on the same day. *β_l* = 6.
β_m The maximum number of clubs based in Greater Manchester which can play at home on the same day. *β_m* = 4.
β_q The maximum number of Premier Division clubs based in London which can play at home on the same day. *β_q* = 3.

Objective Function

$$\text{Minimise} \sum_{i=1}^n \sum_{j=1}^n D_{i,j} x_{i,j} \tag{B.1}$$

Subject to

If a team plays at home on Boxing Day, it must play away on New Year’s Day. This also enforces the opposite condition, *if a team plays at home on New Year’s Day, it must play away on Boxing Day*

$$B_i = 1 - Y_i \quad \forall i \tag{B.2}$$

If team *i* plays team *j* on Boxing Day, then team *j* cannot play team *i* on New Year’s Day

$$\sum_{i=1}^n \sum_{j=1}^n x_{i,j} x_{j,i} = 0 \tag{B.3}$$

Teams which are paired cannot play one another

$$\sum_{i=1}^n \sum_{j=1}^n x_{i,j} P_{i,j} = 0 \tag{B.4}$$

The number of pair clashes for Boxing Day (constraint B.5) or New Year’s Day (constraint B.6) must be less than or equal to the number that appears in the published fixtures.

$$\sum_{i=1}^n \sum_{j=1}^n P_{i,j} B_i \leq Cb_y \tag{B.5}$$

$$\sum_{i=1}^n \sum_{j=1}^n P_{i,j} Y_i \leq Cy_y \tag{B.6}$$

Note: For constraints B.5 and B.6, a pair clash is defined as two paired teams playing at home on the same day (see Table A2).

The maximum number of London-based clubs playing at home on Boxing Day (constraint B.7) or New Year’s Day (constraint B.8) must not exceed a certain threshold.

$$\sum_{i=1}^n L_i B_i \leq \beta_l \tag{B.7}$$

$$\sum_{i=1}^n L_i Y_i \leq \beta_l \tag{B.8}$$

The maximum number of Greater Manchester-based clubs playing at home on Boxing Day (constraint B.9) or New Year’s Day (constraint B.10) must not exceed a certain threshold.

$$\sum_{i=1}^n M_i B_i \leq \beta_m \tag{B.9}$$

$$\sum_{i=1}^n M_i Y_i \leq \beta_m \tag{B.10}$$

The maximum number of London-based Premier Division clubs playing at home on Boxing Day (constraint B.11) or New Year’s Day (constraint B.12) must not exceed a certain threshold.

$$\sum_{i=1}^n Q_i L_i B_i \leq \beta_q \tag{B.11}$$

$$\sum_{i=1}^n Q_i L_i Y_i \leq \beta_q \tag{B.12}$$

Appendix C. Pairings

Table A2 shows the pairings used in this paper. Please note the following:

- If a team is not shown then they are not paired with another team.
- Wimbledon is shown as being paired with Crystal Palace. This is only the case for the 2002–2003 and 2003–2004 seasons. After this, they moved to Milton Keynes (and changed their name to Milton Keynes Dons) and are no longer paired with another team.
- If a team is paired with a team then the arrangement is reciprocal. This fact is reflected in Table A2.

Table A2 Team pairings

<i>Team</i>	<i>Pairs</i>
Arsenal	Tottenham
Aston Villa	Birmingham
Birmingham	Aston Villa
Blackburn	Burnley
Blackpool	Preston North End
Bolton	Stockport County
Bradford City	Huddersfield Town
Brentford	Queens Park Rangers
Bristol City	Bristol Rovers
Bristol Rovers	Bristol City
Burnley	Blackburn
Bury	Rochdale
Cambridge United	Peterborough United
Cardiff City	Swansea City
Charlton	Millwall
Chelsea	Fulham
Chesterfield	Derby County
Colchester United	Ipswich Town
Coventry City	Aston Villa
Crewe Alexandra	Macclesfield Town
Crystal Palace ¹	Wimbledon
Derby County	Chesterfield
Everton	Liverpool
Exeter City	Plymouth Argyle
Fulham	Chelsea
Grimsby Town	Hull City
Hartlepool United	Middlesbrough
Huddersfield Town	Bradford City
Hull City	Grimsby Town
Ipswich Town	Colchester United
Leeds	Bradford City
Leyton Orient	West Ham
Lincoln City	Scunthorpe United
Liverpool	Everton
Luton Town	Watford
Macclesfield Town	Crewe Alexandra
Man City	Man Utd
Man Utd	Man City
Mansfield Town	Nottingham Forest
Middlesbrough	Hartlepool United
Millwall	Charlton
Newcastle	Sunderland
	Coventry City
	Coventry City
	Leeds
	Norwich City
	Birmingham
	Southend United
	Tranmere Rovers
	Leeds
	Southend United
	Huddersfield Town
	Norwich City
	Tranmere Rovers
	Notts. County

Table A2 (continued)

<i>Team</i>	<i>Pairs</i>		
Norwich City	Ipswich Town	Colchester United	Southend United
Nottingham Forest	Mansfield Town	Notts. County	
Notts. County	Mansfield Town	Nottingham Forest	
Oldham Athletic	Wigan Athletic		
Oxford United	Reading		
Peterborough United	Cambridge United		
Plymouth Argyle	Exeter City		
Port Vale	Stoke City		
Portsmouth	Southampton		
Preston North End	Blackpool		
Queens Park Rangers	Brentford		
Reading	Oxford United		
Rochdale	Bury		
Rotherham United	Sheffield Wed	Sheffield United	
Scunthorpe United	Lincoln City		
Sheffield United	Rotherham United	Sheffield Wed	
Sheffield Wed	Rotherham United	Sheffield United	
Southampton	Portsmouth		
Southend United	Ipswich Town	Colchester United	Norwich City
Stockport County	Bolton		
Stoke City	Port Vale		
Sunderland	Newcastle		
Swansea City	Cardiff City		
Swindon Town			
Torquay United			
Tottenham	Arsenal		
Tranmere Rovers	Everton	Liverpool	
Walsall	West Brom	Wolverhampton	
Watford	Luton Town		
West Brom	Walsall	Wolverhampton	
West Ham	Leyton Orient		
Wigan Athletic	Oldham Athletic		
Wimbledon ¹	Crystal Palace		
Wolverhampton	West Brom	Walsall	
Wrexham			
Wycombe Wanderers			
York City			

Appendix D. London- and Manchester-based clubs

Table A3 shows the London- and Manchester-based teams. Please note that Wimbledon was only a London club for the 2002–2003 and 2003–2004 seasons. After this, they moved to Milton Keynes (and changed their name to Milton Keynes Dons) and are no longer considered a London-based team.

Table A3 London- and Greater Manchester-based clubs

<i>Team</i>	<i>London based</i>	<i>Manchester based</i>
Arsenal	X (all)	
Barnet	X	
Bolton		X
Brentford	X	
Bury		X
Charlton	X (all)	
Chelsea	X (all)	
Crystal palace	X (2004–2005)	

Table A3 (Continued)

Fulham	X (all)	
Leyton Orient	X	
Manchester City		X
Manchester United		X
Millwall	X	
Oldham Athletic		X
Queens Park Rangers	X	
Rochdale		X
Stockport County		X
Tottenham Hotspur	X (all)	
West Ham United	X (2002–2003, 2005–2006)	
Wigan Athletic		X
Wimbledon	X	

For the Premier Division London-based clubs, the seasons they were in the Premier Division are shown in brackets.

Received March 2006;
accepted November 2006 after two revisions