Scheduling the English Football League with a Multi-objective Evolutionary Algorithm

Lyndon While[1,⋆] and Graham Kendall[2]

[1] Computer Science & Software Engineering, The University of Western Australia
[2] ASAP Research Group, School of Computer Science, University of Nottingham Malaysia Campus, Malaysia and University of Nottingham, UK
lyndon.while@uwa.edu.au, graham.kendall@nottingham.edu.my

Abstract. We describe a multi-objective evolutionary algorithm that derives schedules for the English Football League over the busy New Year period according to seven objectives. The two principal objectives are to minimise travel distances for teams and supporters, and to minimise so-called "pair clashes" where teams which are geographically close play at home simultaneously, which can cause problems for police, and other logistical issues. The other five objectives implement various problem constraints. The schedules derived are often superior both to those used in the relevant years, and to those previously published in the literature, especially for the harder problem instances. In addition, the system returns a set of schedules offering different trade-offs between the main objectives, any of which might be of interest to the authorities.

Keywords: Sports scheduling, Multi-objective evolutionary algorithms.

1 Introduction

The English Football League is structured around a promotion/relegation system where teams can move up or down divisions, depending on their performances. At the top of the league structure is the English Premier League (EPL), which includes famous teams such as Arsenal, Chelsea, Liverpool, and Manchester United. Beneath the EPL there are three other leagues: the Championship, League 1, and League 2[1]. These four divisions represent the top level of English football.

The EPL has 20 teams and the other three divisions have 24 teams each, a total of 92 teams. Each division is a double round robin tournament, where teams play each other twice, once at each venue. This requires $20 \times 19 = 380$ matches in the EPL, and $24 \times 23 = 552$ matches in each of the others. Thus there are $2,036$ fixtures to be scheduled in any one season, a significant task.

It might be assumed that each division can be scheduled individually, as no team from one division plays against a team from another division. However, dependencies exist between the divisions. The main dependency is around

⋆ Corresponding author.

[1] The actual names change, depending on sponsorship arrangements.

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 842–851, 2014.

pairings. Teams that are paired should (ideally) not play at home on the same day, to reduce the burden on police and other infrastructure. For example, Manchester United and Manchester City are paired as the police would prefer them not to both be playing at home on the same day. Pairings exist across divisions: e.g. Liverpool, Everton, and Tranmere Rovers are all paired.

We focus on scheduling the English Football League over the busy Christmas and New Year period, when each team plays either two or four matches in a short time, and the constraints on the schedule are more difficult than usual, e.g. bad weather, police leave, and clashes with other holiday commitments for supporters. We try to minimise the distance traveled by supporters, which was the primary focus of our previous work[1,2,3]. We would also like to minimise the number of *pair clashes* where paired teams play at home on the same day. Minimising pair clashes and minimising travel are conflicting objectives and it seems appropriate to investigate a multi-objective approach for this problem. Previous attempts[3] have used multiple runs of a single-objective algorithm.

The principal contribution of this paper is the description and analysis of a multi-objective evolutionary algorithm that derives schedules over the Christmas and New Year period. The algorithm minimises the total travel distance (to save supporters from having long journeys), and it also minimises the number of pair clashes (to save policing resource). Five constraint objectives are also managed: these are presented in Section 3, and a more formal problem description is given in [1]. The schedules presented are superior both to those used by the football authorities, and also to those previously published in the literature, especially for the larger problem instances. In addition, the system returns a set of schedules offering different trade-offs between the two objectives, any of which might be of interest to the authorities.

The rest of the paper is structured as follows. Section 2 describes relevant background material and previous work, and Section 3 describes the problem that we address. Section 4 describes the details of our multi-objective approach, and Section 5 describes the results achieved. Section 6 concludes the paper.

2 Background

2.1 Multi-objective Approaches to Sports Scheduling

Multi-objective approaches to sports scheduling have received limited attention in the literature. This is perhaps surprising, given the multi-objective nature of this sector, e.g. maximise crowd attendance, minimise costs, minimise policing, maximise sales, minimise travel, maximise the gap between return matches, etc.

Evans[5], as far back as 1988, proposes a decision support system for scheduling umpires in baseball. One objective is to minimise the travel costs of umpires, with the other objective being concerned with the number of times an umpire can officiate over a given team.

Duarte and Ribeiro[4] consider a bi-objective referee assignment problem, where they minimise the difference between the target and assigned games for each referee, and minimise the idle time between games for each referee.

Barone *et al.*[6] investigate the Australian Football League wrt a number of competing objectives: balancing the number of home games for each team, balancing the sequence of home/away games (as too many consecutive games at home (resp. away) is viewed as unfair), balancing the effects of travel in a large country, maximising profit, and distributing the games across the country on a given weekend. Their multi-objective evolutionary algorithm was able to produce good results which offer a range of options to the client.

While and Barone[7] focus on Super 14 rugby. This tournament takes place in several countries (Australia, New Zealand, and South Africa), distributed over twelve time zones. They balance the number of home games a team has, whilst optimising travel requirements and distributing the games in each round evenly across the three countries, to optimise the use of prime time TV spots. Again they use a multi-objective evolutionary algorithm, producing results that offer significant improvements over the fixtures adopted by the organising body.

Craig *et al.*[8] use a multi-objective evolutionary algorithm to investigate the National Hockey League. The objectives are minimising travel, providing equity in rest time between games, and minimising consecutive home/away game sequences. The results are superior to the 2008–09 schedules that were used, and they offer a range of trade-offs due to the nature of the multi-objective algorithm.

Kendall and Westphal[2] use CPLEX in an attempt to solve the problem of scheduling football matches over the New Year period whilst trying to minimise the total distance travelled. From a base model they vary the parameters in order to find solutions more acceptable to the individual clubs, e.g. by limiting the maximum distance that a single team would travel. A variety of experiments are presented to demonstrate the effectiveness of this approach.

Kendall *et al.*[3] again explore the idea of using multiple runs with varying parameters. They minimise both the distance travelled and the number of pair clashes in an attempt to find good trade-offs. The current paper tackles the same problem as [3], but using a multi-objective approach. The aim is to get better solutions in a single run, both saving time and offering the client superior solutions.

Survey papers related to sports scheduling are available in [9,10].

2.2 Multi-objective Optimisation

In a multi-objective optimisation problem, potential solutions are assessed according to two or more independent quantities. The characteristic of good solutions is that improving in one objective can be achieved only by worsening in at least one other objective. An algorithm for solving such problems returns a set of solutions offering different trade-offs between the various objectives.

Consider a problem where the fitness function maps a solution x into a fitness vector $\overline{f_x}$. A solution x *dominates* a solution y iff $\overline{f_x}$ is at least as good as $\overline{f_y}$ in every objective, and is better in at least one objective. x is *non-dominated* wrt a set of solutions X iff there is no solution in X that dominates x. X is a *non-dominated set* iff every solution in X is non-dominated wrt X. The set of fitness vectors corresponding to a non-dominated set is a *non-dominated front.*

A solution x is *Pareto optimal* iff x is non-dominated wrt the set of all possible solutions. Such a solution is characterised by the fact that improvement in one objective comes only at the expense of other objectives. The *Pareto optimal set* is the set of all Pareto optimal solutions. Multi-objective optimisation aims to find (or approximate) this Pareto optimal set.

With multiple objectives there is only a partial order on solutions, which causes problems for selection in an evolutionary algorithm. The usual solution is to define a ranking on solutions: one popular scheme[11] defines the *rank* of a solution x wrt a set X to be the number of solutions in X that dominate x. Selection is then based on ranks: a lower rank implies a better solution.

Precise definitions of all these terms can be found in [12].

3 Problem Statement

As stated in Section 1, the top four English divisions contain 92 teams, and require 2,036 fixtures to be scheduled each season. Over Christmas and New Year, each team plays either two or four matches. One (resp. two) match has to be played at home, and the other (resp. two) has to be played at an away venue. This time of the year is one of the few times when all teams have to play. It may seem that scheduling part of a season might just create difficulties for other dates, but in these competitions many games are moved each year for a variety of reasons, so it is common for the authorities to focus their efforts in this way.

The full problem can be defined as follows. Further details are available in [1]. Given a set of teams allocated to the four divisions in a given year, given the distance between each pair of teams, and given for each team a set of "pairs":

1. There are either two or four rounds (this depends on the season, but following [2] we always generate a two-round fixture and a four-round fixture).
2. Each team plays one game in each round against another team in its division.
3. Each team alternates home and away games.
4. No team can play another team more than once.
5. No team can play against any of its pairs (this constraint is sometimes violated by the football authorities, but we enforce it).
6. Only six London teams can play at home in any round.
7. Only three London EPL teams can play at home in any round.
8. Only four Greater Manchester teams can play at home in any round.

A schedule is measured against two objectives:

- The total distance travelled by the away teams across all rounds.
- The total number of pair clashes (where paired teams play at home on the same day) across all rounds. For the first six seasons we study, the minimum possible clashes for the two-round case is eight[2]. In 2008–09, Mansfield Town were demoted from the league, reducing the minimum to seven.

[2] Three mutually-clashing teams implies at least one clash/two rounds (six occurrences of this), four mutually-clashing teams implies at least two clashes/two rounds (one occurrence of this). Note that the number of pair clashes for four rounds is doubled, because the home teams in Round k are the same in Round $k+2$.

For each scenario, a target number of clashes is given based on the schedule actually used in that year[1,2].

4 Methodology

This section describes the details of our algorithm: the genetic representation and variation operators used, the seven objectives and their quantification, and details of initialisation, termination, and archiving.

4.1 Representation

Given a problem instance with k rounds, for a division with n teams we have

- a permutation of the teams in that division;
- $k-1$ permutations of the numbers $0 \dots n/2-1$.

The translation from genotype to phenotype (i.e. the schedule) is best explained via a worked example. With four rounds and a single division containing the six teams `A,B,C,D,E,F`, the genotype

```
EDAFBC - 201 - 120 - 021
```

represents the schedule

```
Round 1: E vs F, D vs B, A vs C
Round 2: F vs A, B vs E, C vs D
Round 3: E vs B, D vs C, A vs F
Round 4: F vs E, B vs A, C vs D
```

In Rounds 1 and 3, the teams in the first half of the main permutation (i.e. `E, D, A`) play at home: in Rounds 2 and 4, the teams in the second half of the main permutation (i.e. `F, B, C`) play at home. In Round 1, the away teams are allocated directly from the second half of the permutation: in Round $j+1$, the j^{th} list of indices is used to permute the away teams.

Note that the genotype for a particular problem instance will contain one of these combinations for each division in the league. For representation purposes, the divisions can be treated entirely separately.

The representation enforces directly the number of rounds scheduled, and the requirements that each team plays once per round and that teams alternate home and away games (i.e. 1–3 from Section 3). The variation operators described in Section 4.2 maintain these requirements. The other requirements (i.e. 4–8 from Section 3) are enforced via so-called "constraint objectives" in the fitness calculations.

4.2 Variation Operators

Mutation involves randomly selecting either 1 or 2 of the permutations in the genotype, and in each one separately, swapping two of its entries. This mutation is guaranteed to maintain feasibility in the genotype.

No crossover is currently used.

4.3 Objectives

We implement seven objectives. The first five are constraint objectives where a value that is too high indicates an infeasible solution[13,14,15].

1. **Repeats**: Across all rounds, we count the number of games that are repeated (i.e. where t_i plays t_j twice): none are allowed.
2. **Derbies**: Across all rounds, we count the number of games between paired teams: none are allowed.
3. **London EPL**: In each round, we count the number of EPL games played in London: we allow three/round.
4. **London**: In each round, we count the number of games played in London: we allow six/round.
5. **GMR**: In each round, we count the number of games played in Greater Manchester: we allow four/round.
6. **Pair clashes**: In each round, we count the number of times that a team and one of its pairs are both playing at home, and we sum these values for all rounds: we allow a total that depends on the year, from Table 33 of [2].
7. **Distance**: Across all rounds, we sum the distances travelled by the away teams in all games.

Note that the limits set for London EPL, London, and GMR are the minimum values possible given the number of teams in each region and division.

We have tried various ways of combining the five constraint objectives, but we have found that maintaining them as separate objectives gives the best diversity in the population.

4.4 Other Algorithm Details

Selection: We use standard Pareto ranking[11] in selection, and where we need to break ranks, we favour the constraint objectives over the "real" objectives, in the order described in Section 4.3. Thus feasible solutions are favoured in selection, which will tend to increase the proportion of such solutions over time.

Archiving: All non-dominated feasible solutions with distinct fitnesses are archived and are presented in the results of the algorithm. This approach works well because of the discrete nature of pair-clash-counts: the archive will contain only a small part of the population, never more than about twenty solutions.

Initialisation: For the initial set of solutions, we simply choose each permutation randomly.

Termination: The algorithm runs until the feasible solution with minimum pair clashes ceases to show improvement. For the experiments described in Section 5, this typically took 10–20,000 generations.

5 Results

We tested our algorithm on fourteen problem instances. We used the seven seasons 2002–03 to 2008–09 inclusive, and for each season we ran the algorithm to

Table 1. For the 2- and 4-round cases, we give the lower bounds from [2]; the distances for [2] and for the current algorithm; the percentage differences between them; and the run-times. Negative percentages indicate improvements for our proposed algorithm. Each result is from one run with a population of 1,500, roughly 10–20,000 generations.

Two rounds					
	bound	[2]	current	improv.	minutes
2002–3	4,692	4,801	4,825	0.5%	533
2003–4	5,186	5,209	5,244	0.7%	500
2004–5	5,135	5,161	5,216	1.1%	263
2005–6	5,038	5,038	5,067	0.6%	398
2006–7	5,295	5,308	5,374	1.2%	272
2007–8	5,020	5,034	5,076	0.8%	180
2008–9	5,243	5,244	5,259	0.3%	190
Four rounds					
	bound	[2]	current	improv.	minutes
2002–3	11,377	13,813	13,773	–0.3%	643
2003–4	11,896	13,966	13,870	–0.7%	483
2004–5	12,040	13,605	13,576	–0.2%	298
2005–6	12,221	13,785	13,612	–1.3%	328
2006–7	12,409	14,262	14,076	–1.3%	652
2007–8	11,985	14,089	13,897	–1.4%	440
2008–9	12,283	14,671	14,529	–1.0%	501

produce two and four rounds of fixtures. This is consistent with [2] and enables us to compare our results directly. We also compare results with [3], but this paper produced only two rounds of fixtures.

Table 1 summarises the best distance results achieved for results that respect all of the problem constraints, including the same limits on pair clashes as used in [2]. Our runtimes are in line with those used in [2], accounting for the different computer configurations and bearing in mind that [2] did many runs for each instance, whereas our multi-objective approach requires only a single run. Clearly if this methodology were used in the real world, the run times would be acceptable given that it is a once-a-year operation and that the schedules are important from many perspectives, including financial.

It is clear that our multi-objective algorithm performs slightly worse than [2] on the 2-round fixtures where they get very close to the optima, and slightly better on the more complex 4-round fixtures. It is fairly typical of evolutionary approaches to perform (relatively) better on harder problems.

Figure 1 shows the complete results on the 2-round fixtures. The y-axis shows the difference to the distances reported for [2] in Table 1, and the x-axis shows the number of pair clashes, relative to the numbers that were used in the fixtures that were published by the football authorities. That is, a value of zero means that we have the same number of pair clashes, and a negative (resp. positive) value means that we have generated a schedule with fewer (resp. more) pair clashes than the published fixture. Thus the result for [2] sits at $(0, 0)$ for every line. It is not surprising that, as we allow more pair clashes, we have less distance

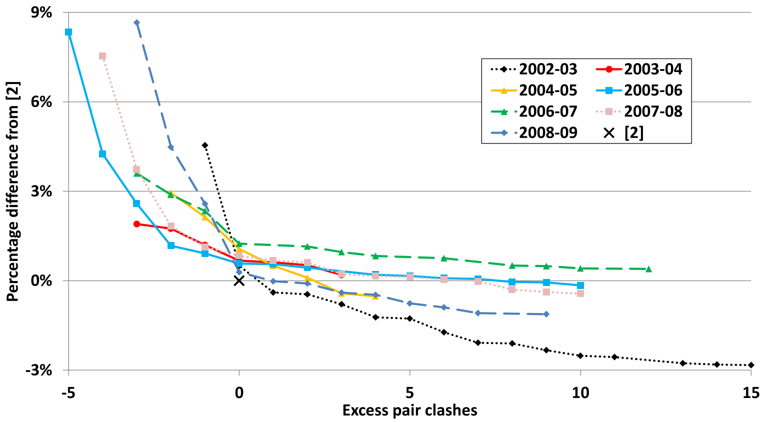

Fig. 1. For the 2-round case, each line plots the best distance result achieved with the corresponding number of pair clashes in excess of the [2] limits, for the same runs as Table 1 plus another set of runs that tried to minimise clashes as far as possible. Negative values indicate reduced distances travelled. The leftmost point on every line has the smallest number of clashes possible, usually eight (but seven for 2008–09).

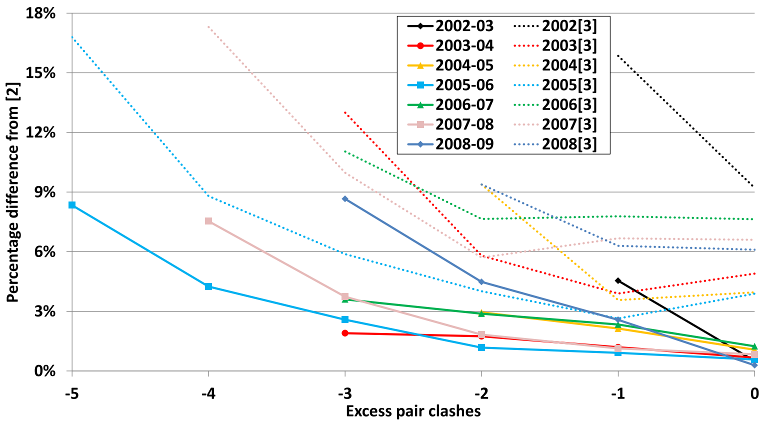

Fig. 2. For the 2-round case, each solid line plots the same data as Figure 1 (in the relevant range of pair clashes), and each dotted line plots the best results from [3]. The leftmost point on every solid line has the smallest number of clashes possible.

to travel. In 2002-03, for example, if we allow fifteen extra pair clashes, we can save almost 3% on the distance travelled.

Figure 2 shows a subset of the data from Figure 1, but compared with the multi-objective data from [3]. The distinct advantage of the new approach is clear: it produces results that are uniformly better, often by 5–10%.

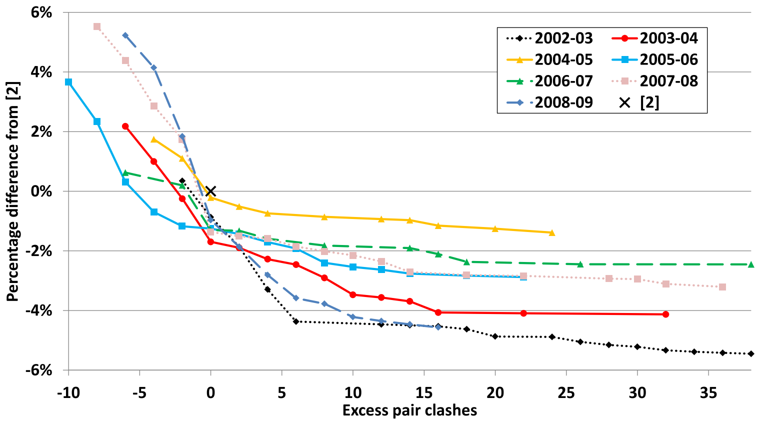

Fig. 3. For the 4-round case, each line plots the best distance result achieved with the corresponding number of pair clashes in excess of the [2] limits, for the same runs as Table 1 plus another set of runs that tried to minimise clashes as far as possible. Negative values indicate reduced distances travelled. The leftmost point on every line has the smallest number of clashes possible, usually sixteen (but fourteen for 2008–09).

Figure 3 shows the complete results on the 4-round fixtures. The overall pattern is the same: as we allow more pair clashes, we reduce the overall distance that has to be travelled. It is noticeable that for the same number of pair clashes (i.e. zero on the graph), the multi-objective approach is able to produce shorter overall travel distances than [2]: occasionally it can also do this for fewer pair clashes (i.e. negative on the graph). Given that the multi-objective approach is also able to offer a set of potential solutions offering varying trade-offs, this is a major advantage of this methodology.

6 Conclusions

We have described a multi-objective evolutionary algorithm that derives schedules for the busy New Year period in the English Football League. The system minimises both away teams' travel and "pair clashes" where teams which are geographically close play at home on the same day, whilst also managing five constraint objectives which restrict the types of games allowed and the numbers of games allowed in London and in Greater Manchester. The schedules derived are superior to those previously published, especially for larger problem instances: also they offer a range of different trade-offs between the main objectives.

Future work will extend the system to incorporate fairness, as defined in [2]: enforcing a maximum distance for individual teams, additional to the existing objectives. We will also explore enhanced features such as weighted pairings, where clashes between "big teams" are viewed as worse than other clashes.

References

1. Kendall, G.: Scheduling English Football Fixtures over Holiday Periods. JORS 59(6), 743–755 (2008)
2. Kendall, G., Westphal, S.: Sports Scheduling: Minimizing Travel for English Football Supporters. In: Uyar, A.S., Ozcan, E., Urquhart, N. (eds.) Automated Scheduling and Planning. SCI, vol. 505, pp. 61–90. Springer, Heidelberg (2013)
3. Kendall, G., McCollum, B., Cruz, F.R.B., McMullan, P., While, L.: Scheduling English Football Fixtures: Consideration of Two Conflicting Objectives. In: Talbi, E.-G. (ed.) Hybrid Metaheuristics. SCI, vol. 434, pp. 369–385. Springer, Heidelberg (2013)
4. Duarte, A.R., Ribeiro, C.C.: Referee Assignment in Sports Leagues: Approximate and Exact Multi-objective Approaches. In: 19th International Conference on Multiple Criteria Decision Making, Auckland, New Zealand, pp. 58–60 (2008)
5. Evans, J.R.: A Microcomputer-based Decision Support System for Scheduling Umpires in the American Baseball League. Interfaces 18(6), 42–51 (1988)
6. Barone, L., While, L., Hughes, P., Hingston, P.: Fixture Scheduling for Australian Rules Football using a Multi-objective Evolutionary Algorithm. In: IEEE CEC, Vancouver, Canada, pp. 3377–3384 (2006)
7. While, L., Barone, L.: Super 14 Rugby Fixture Scheduling using a Multi-objective Evolutionary Algorithm. In: IEEE CI-Scheduling, Honolulu, Hawaii, pp. 35–42 (2007)
8. Craig, S., While, L., Barone, L.: Scheduling for the National Hockey League using a Multi-objective Evolutionary Algorithm. In: Nicholson, A., Li, X. (eds.) AI 2009. LNCS, vol. 5866, pp. 381–390. Springer, Heidelberg (2009)
9. Kendall, G., Knust, S., Ribeiro, C.C., Urrutia, S.: Scheduling in Sports: an Annotated Bibliography. Computers & Operations Research 37(1), 1–19 (2010)
10. Rasmussen, R.V., Trick, M.A.: Round Robin Scheduling — a Survey. EJOR 188(3), 617–636 (2008)
11. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: ICGA, Urbana-Champaign, Illinois, pp. 416–423 (1993)
12. Coello Coello, C.A., Lamont, G., Van Veldhuizen, D.: Evolutionary Algorithms for Solving Multi-objective Problems. Springer (2007)
13. Hingston, P., Barone, L., Huband, S., While, L.: Multi-level Ranking for Constrained Multi-objective Evolutionary Optimisation. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 563–572. Springer, Heidelberg (2006)
14. Huband, S., Tuppurainen, D., While, L., Barone, L., Hingston, P., Bearman, R.: Maximising Overall Value in Plant Design. Minerals Eng. 19(15), 1470–1478 (2006)
15. While, L., Hingston, P.: Usefulness of Infeasible Solutions in Evolutionary Search: an Empirical and Mathematical Study. In: IEEE CEC, Mexico, pp. 1363–1370 (2013)