Problem formulation and solution for a real-world sports scheduling problem

MIKE B. WRIGHT
Department of Management Science
Lancaster University
LA1 4YX
U.K.
http://www.lums.lancs.ac.uk/mansci/profiles/76/

Abstract: - This paper describes a problem faced every year by the Devon Cricket League in England. Every league match requires two officiating umpires. There are various preferences relating to the deployment of umpires which fall short of being definite constraints. The requirement was to produce a computer system which would allocate umpires in a satisfactory manner without the need for human interaction, since the users of the system will be unqualified amateurs. This necessitated the construction of a very complex objective function, using imprecise and ill-defined information. The biggest challenge was thus one of problem formulation; the solution technique was relatively straightforward. The system has been put into successful use and it is expected that its use will continue indefinitely.

Key-Words: - Scheduling, timetabling, sport, cricket, problem formulation, multiple objectives

1 Introduction

1.1 Scheduling of sports officials

There have been many papers in the academic literature concerned with the scheduling of sports fixtures. Some of these have been theoretical in nature [2], [6], [7], [8], while others have focused upon case studies [1], [5], [9], [13]..

However, papers concerned with the scheduling of sports officials are very few and far between. Apart from [3], [4] and [10] there appears to be nothing. This is surprising given that it must be an important issue for all sports leagues and tournaments world-wide.

This paper concerns the scheduling of umpires for the Devon Cricket League, an amateur league in England. It concentrates on the problem formulation and shows that a significant amount of interaction with the model user is necessary in order to come up with a fully satisfactory objective function which incorporates all the practical complexities. This was the most challenging part of the work; the solution technique was straightforward in comparison.

This forms an interesting contrast with most academic research in the area, which tends to assume that formulation is straightforward and thus concentrates on refining the solution techniques used.

1.2 The Devon Cricket League

The Devon Cricket League is an amateur league which spans one of the largest counties in England. Matches occur on Saturday afternoons from late April or early May until late August or early September.

Although the League is currently divided into eight divisions, the umpire allocation exercise described here involves only the top three divisions; umpires for the other divisions are sorted out later on a more local basis. Each division contains ten amateur clubs, each of which has its own home ground and plays fixtures against every other club in its division exactly twice, once at home and once away. The fixtures are arranged so that one of these matches takes place in the first half of the season, and one in the second half. Every club in the top three divisions has a match every Saturday during the season, which lasts eighteen weeks.

Each match requires two umpires who are paid a small fee as well as travel expenses. As the league is run on a financial shoestring, it is of vital importance to keep these expenses low; however, there are also several other considerations to take into account when deciding which umpires should be allocated to each match. Thus the problem of allocating umpires to matches after they have been timetabled is decidedly non-trivial.

This task falls to the league secretary – an unpaid role usually occupied by a retired cricket player who is unlikely to be a sophisticated computer user. The current occupant of this role therefore appealed for help in allocating the League's umpires.

The umpire allocations are derived in two separate tranches, one for the first half of the season and one for the second. The timing of these tranches is under the control of the league secretary. The first tranche needs to take place a few weeks in advance of the start of the season. The timing of the second tranche depends on two conflicting factors: the need to give advance information to umpires and the desirability of taking account of unforeseen events during the first half of the season.

2 Problem Formulation

2.1 Factual information required

The model requires a susbstantial amount of detailed factual information regarding the clubs, the matches and the umpires, all of which may change substantially from year to year and must therefore be in a format that can conveniently be entered and amended by the system user.

For each club, the data to be input is its name, its address (in the form of geographical co-ordinates) and the division in which it is playing. For each match, the system requires the home team, the away team and the date.

Umpire information is a little more complex. The league has a pool of around fifty umpires, who all live within the county of Devon or in a neighbouring county (Cornwall, Dorset or Somerset), near to one of Devon's borders. Their names and home addresses (again in co-ordinate form) must all be entered.

Many umpires are not available every Saturday of the season – indeed, some may only be available for the occasional date – and therefore full availability information must be entered.

Moreover, not all umpires are equal. Each umpire is allocated a rating (or status) of 1, 2, 3 or 4 by the league, reflecting his quality and experience. This affects the matches at which he (they are all men) is allowed to officiate (see later).

When allocating umpires for the second half of the season, the first half's allocations are also used as input, but may need to be changed manually before running the system to reflect any emergency changes that were made (e.g. because of illness).

2.2 Preference information

In addition to the factual data, the system requires information regarding preferences as to how the umpires should be allocated. Some of these are assumed by the program; others are entered by the user. Some of the latter may override some of the former.

None of these preferences is treated by the system as a constraint – indeed, it is usually the case that several combinations of preferences are mutually incompatible – but as part of the objective function, in that penalty costs will be applied when they are not adhered to. These penalty costs can be varied by the user.

2.2.1 Automatic preferences

The automatic preferences were determined after discussions of general principles with the user of the system. They are:

- no umpire to officiate at a match involving a particular club more than once in each half of the season
- no umpire to officiate at two matches involving a particular club less than six weeks apart
- no umpire to officiate at a match involving a particular club *as home team* more than once in the entire season
- no umpire to officiate at two matches involving a particular club *as home team* less than eight weeks apart (this of course can happen only if the previous preference is broken)
- no pair of umpires to officiate together more than once during the entire season
- no pair of umpires to officiate together at two matches less than eight weeks apart (again, this can happen only if the previous preference is broken)
- no umpire may officiate in two matches both of which involve the same two clubs
- no pair of umpires to officiate together in a match for which they are *both* of a higher status than necessary

The first seven of these are required because it is deemed usually desirable to spread umpires between clubs, home grounds and other umpires. The eighth is needed because it is thought wasteful of the best umpires; sometimes it is helpful for a top umpire to officiate in a low division in order to help an inexperienced colleague, but there is normally no point in having two top umpires at a lowly-rated match.

The precise number of weeks between matches specified in three of these preferences was determined after some discussion with the user, whose initial requirement was that there should be "a reasonable gap". These are examples of where an ill-defined but important requirement has to be defined precisely in quantitative terms in order that it can be taken into account by the program. Inevitably this implies an element of arbitrariness in the formulation, but there is no practical alternative.

For the first six of these automatic preferences, the penalty cost of a broken preference increases with the amount by which the preference is broken. Moreover, this increase follows a squared rather than a linear function – thus, for example, if a pair of umpires is together three times, the penalty cost will be four times the cost which would apply if they were together only twice. This is because it is usually more acceptable to allow several occasions where a preference is just missed than a relatively small number of occasions where it is missed by a larger amount.

2.2.2 User-defined preferences

One set of preferences that must be entered by the user lists the target number of matches for each umpire, overall and in each division, in each half of the season. This can be a painstaking business, since it is advisable to ensure that the totals are correct and that the combination of these targets and the umpire availability details can lead to a feasible solution. A separate routine is available for checking this.

Another essential set of preferences defines the effect of the umpire status information. The simpler form of this preference simply states that umpires of status x should not be used for matches in division z. In its more complex form, the preference says that a pair of umpires with statuses $\{x, y\}$ should not officiate at a match in division z.

Another type of preference says that umpire x must, or must not, be allocated to a specific match m. Alternatively it may be specified that an umpire must have a match (without specifying which one) on a given date. Slightly more complex is a preference that says that matches m and m' must not have an umpire in common.

Other user-defined preferences involve specific amendments to the automatic preferences, which can have the effect of either strengthening or weakening them. For example:

- A specific umpire should not officiate at any match involving a specific club (e.g. because he used to be a player for that club)
- A specific pair of umpires may not officiate together at all (e.g. because they have a history of incompatibility)
- A specific pair of umpires should officiate together three times during the season (it can be a requirement for the full qualification of a junior umpire that he officiates together with a specific senior umpire for training purposes)
- An umpire may make a specific request, for example that he has at least one match at a particular ground, which the user chooses to aim to satisfy.

Penalty costs for breaking these preferences have the same squared structure as for automatic preferences.

2.3 Travel costs

The clubs' grounds and umpires' homes are all located within a 110 kilometre square area. Precise locations are represented using a 10 kilometre square grid within this area. Thus the west-east co-ordinate varies between 1 and 11 and the south-north co-ordinate likewise varies between 1 and 11. Pythagorean distances are used; thus, for example, the travel distance for an umpire living in square [2,9] to officiate at a ground located at [8,4] is approximated as $\sqrt{(8-2)^2 + (9-4)^2} =$ 7.8 units (i.e. 78 kilometres).

This formulation, while simple and generally adequate, is somewhat approximate and in future a rather more accurate method of estimating distances may be incorporated.

However, an important consideration is not just the total travel distance but the number of long journeys any single umpire must undertake, since it is desirable to spread the long journeys around, rather than overloading just a few umpires with several such journeys (though in practice any umpire living on the fringes of the county will have to travel more than those living near the centre). Therefore a definition of a "long" journey is required. This is currently set at 5.5 units (55 kilometres).

It is also common practice, if the two umpires allocated to a match both have to travel a long way, for one of them to be encouraged to give the other a lift if this does not involve too large a detour. Thus, within the computer system, a shared journey is defined as one where both umpires live a long way (i.e. more than 55 kilometres) away from the match location and where the umpire giving the lift will not by doing so increase his journey length by more than 25%. This exact definition is rather arbitrary and can be changed by the user if required.

The travel cost function for an umpire thus consists of two elements: one proportional to the total distance driven (including detours required to pick up another umpire, but excluding journeys where the umpire is a passenger in another umpire's car), one a penalty cost (with a square function as before) if the number of long journeys (whether as driver or as passenger) is greater than two in either half of the season.

2.4 Cost of change

It can sometimes happen that a set of allocations is published, but then needs to be changed because of altered circumstances. If the changes required are only very small, the user may be happy to make such changes manually, but if they are at all substantial it is very useful to be able to take an existing solution as the initial solution and work from there.

In such cases, it is desirable that there should not be too many changes from the original schedule, and hence a cost of change is required; every umpire allocation that differs from the original allocation incurs this cost.

The precise value of this cost may be quite critical; too high and nothing changes, too low and far too much changes. The user will therefore probably try a number of alternative values before finding one that produces the amount of change required.

2.5 Cost weightings

Since it is exceptionally unlikely that a cost-free solution can be provided, the relative weights of all the different elements of cost must be chosen with care. To some extent they can be set in advance with reference to user priorities, but initially a great deal of trial and error was involved, punctuated by frequent discussions with the user as to which of a number of situations was preferred.

Normally the user-defined preferences will carry a heavier weight than the automatic preferences, and it would normally be expected that the vast majority would be satisfied. However, none of them is regarded as a binding constraint.

The setting of weights introduces further arbitrariness into the problem formulation, but again there is no practical alternative. All the factors specified need to be included in the model if the user is to be satisfied with the solutions produced.

2.6 Constraints

The only factors regarded as binding constraints are that no umpire may officiate at more than one match on any given date, or on any date for which he is unavailable. Otherwise, any allocation is regarded as being feasible.

3 Problem Solution

3.1 Solution technique

This problem formulation clearly cannot be solved to optimality because of its size, complexity and non-linearity. In any case, the whole concept of optimality is not at all well-defined because of the inevitable arbitrariness involved in parts of the cost function.

However, the formulation lends itself naturally to being solved by a neighbourhood search metaheuristic. This requires the specification of the following [14]:

Search space definition: every feasible solution is regarded as a point in search space.

Generation of an initial solution: this is obtained by means of a quick constructive method which makes feasible allocations at random manner. It may happen towards the end of this process that no feasible allocation exists for a match. In this case, a "dummy" umpire is allocated, which incurs a large "absence" cost.

Neighbourhood definition: given a current solution, a neighbouring solution is one that can be reached either by swapping a pair of umpires between two matches or by replacing one umpire by another for a single match. Where the replaced umpire is a dummy umpire, this will always result in an overall cost decrease.

Neighbour generation: at each step, a neighbouring solution is chosen entirely at random.

Acceptance criterion: this is the criterion used in a variation of Simulated Annealing known as *Subcost-Guided Simulated Annealing* (SGSA) [11], [12]. This is the same as standard simulated annealing (SA), except that the cost increase is adjusted before being used in the acceptance criterion.

The acceptance criterion for SGSA is that a worsening perturbation is accepted if $\mathbf{R} < \mathbf{e}^{-(\mathbf{C'}/\mathbf{T})}$, where **R** is a random number between 0 and 1, **T** is the temperature and **C'** is an adjusted cost increase, defined as $\mathbf{C'} = \mathbf{Ce}^{-\theta\mathbf{B}/\mathbf{C}}$, where **C** is the overall cost increase, **B** is the best decrease for any individual subcost and θ is a parameter. As with SA, a non-worsening neighbour is automatically accepted.

The SGSA criterion thus requires three parameters – the start temperature, end temperature and θ. Default values are used which relate to the default cost weightings – the user is not expected to have sufficient understanding to vary these.

Stopping criterion: the search continues for a predetermined number of iterations, whose default value is currently set in order that the user may run the program on his computer (an ordinary PC) to reach a solution in about ten minutes.

This technique was chosen because it proved to give slightly better results during tests than standard SA, as long as the parameters were chosen carefully. However, the advantage is relatively small, and even a simple local improvement technique frequently produces solutions of a quality that the user would be very happy to accept.

It therefore appears that, for this instance at any rate, the choice of solution technique is not nearly as important as the careful formulation of the objective function.

3.2 Results

The results file for a solution includes:

- A list of all matches, ordered by division and date, with names of umpires added
- A chronological list of matches, dates and partner umpires for each umpire, together with details of where it is assumed that the umpires will travel together
- a table summarising the number of matches in each division for each umpire
- a table summarising the activity assigned to each umpire on each date
- a matrix showing umpire-club incidence
- a matrix showing umpire-ground incidence
- a matrix showing umpire-umpire incidence

However, the system can be set up to produce any number of solutions. If more than one is requested, the results file names are presented to the user in ascending order of cost. The expectation is for the user to look at solutions in this order until one is found that is satisfactory, though on every occasion to date in practice the user has been happy to accept the first solution on the list.

4 Conclusion

4.1 Use of the system

At the time of writing, the system has been used for the Devon Cricket League for both halves of the 2003 and 2004 seasons and the first half of 2005. The Devon Cricket League secretary has written:

"We now have a more than satisfactory set of appointments. As the only people within the Devon League that knew of the programme's existence, the subcommittee casually asked various umpires what they thought of the season 2003 appointments. To our delight they thought it was the best list produced for many years as they were getting a fairer selection of games and a wider variety. Those going for full membership were all happy, as they had got what they had requested to enable them to apply for their full membership of the ACU&S (*the Association of Cricket Umpires and Scorers)*. The outcome of all this was that not only were the appointed umpires in the right place at the right time but were getting a fairer distribution of games as well as complying with League and ECB requirements. We now have umpires requesting to join the panel. To our subcommittee this has saved us many man hours of work, thus releasing us for other tasks which we were previously unable to do."

The full data files used in any of 2003, 2004 or 2005 are available on application to the author.

4.2 Implications for other research

This case-study highlights the importance of thorough and detailed problem formulation for real-world complex problems, with frequent reference to the future user in order to ensure that all relevant considerations are included in the model. The approach used here could be used for a very wide variety of types of real-world application where there are many types of preference to consider.

In these circumstances, it will often be the case that problem formulation is the more difficult and the more important part of the work. The choice of solution technique, though not wholly unimportant, may well be a secondary consideration.

References:

[1] della Croce, F., Tadei, R. and Asioli, P.S., Scheduling a round robin tennis tournament under courts and players availability constraints, *Annals of Operations Research,* Vol. 92, 1999, pp. 349-361.

[2] Easton, K, Nemhauser, G.L. and Trick, M.A., Solving the travelling tournament problem: a combined integer programming and constraint programming approach, *Lecture Notes in Computer Science,* Vol. 2740, 2003, pp. 10-109.

[3] Evans, J.R., Hebert, J.E. and Deckro, R.F., Play ball – the scheduling of sports officials, *Perspectives in Computing,* Vol. 4, No. 1, 1984, pp. 18-29.

[4] Evans, J.R., A microcomputer-based decision support system for scheduling umpires in the American baseball league, *Interfaces,* Vol. 18, No. 6, 1988, pp. 42-51.

[5] Nemhauser, G.L. and Trick, M.A., Scheduling a major college basketball conference, *Operations Research,* Vol. 46, No. 1, 1998, pp. 1-8.

[6] Schonberger, J., Mattfeld, D.C. and Kopfer, H., Memetic algorithm timetabling for non-commercial sport leagues, *European Journal of Operational Research,* Vol. 153, No. 1, 2004, pp. 102-116.

[7] Trick, M.A., Integer and constraint programming approaches for round-robin tournament scheduling, *Lecture Notes in Computer Science,* Vol. 2740, 2003, pp. 63-77.

[8] Urban, T.L. and Russell, R.A., Scheduling sports competitions on multiple venues, *European Journal of Operational Research,* Vol. 148, No. 2, 2003, pp. 302-311.

[9] Willis, R.J. and Terrill, B.J., Scheduling the Australian state cricket season using simulated annealing, *Journal of the Operational Research Society,* Vol. 45, No. 3, 1994, pp. 276-280.

[10] Wright, M.B., Scheduling English cricket umpires, *Journal of the Operational Research Society,* Vol. 42, No. 6, 1991, pp. 447-452.

[11] Wright, M.B., Subcost-Guided Search – experiments with timetabling problems, *Journal of Heuristics,* Vol. 7, No. 3, 2001, pp. 251-260.

[12] Wright, M.B. Subcost-Guided Simulated Annealing. In: Ribeiro C.C. and Hansen P. (Eds.), *Essays and surveys in metaheuristics,* Kluwer Academic Publishers, Boston, 2001, pp. 631-639.

[13] Wright, M.B., Scheduling fixtures for Basketball New Zealand, *Lancaster University Management School Working Paper*, Lancaster, LUMSWP2003/039, 2003, to appear in *Computers and Operations Research*.