

Edith Cowan University
Research Online

ECU Publications Pre. 2011

2006

Fixture-scheduling for the Australian Football League using a Multi-objective Evolutionary Algorithm

Luigi Barone
University of Western Australia

Lyndon While
University of Western Australia

Paul Hughes
University of Western Australia

Philip Hingston
Edith Cowan University

Follow this and additional works at: <https://ro.ecu.edu.au/ecuworks>

 Part of the [Computer Sciences Commons](#)

[10.1109/CEC.2006.1688413](https://ro.ecu.edu.au/ecuworks/2025/10.1109/CEC.2006.1688413)

This is an Author's Accepted Manuscript of: Barone, L., While, L., Hughes, P., & Hingston, P. F. (2006). Fixture-scheduling for the Australian Football League using a Multi-objective Evolutionary Algorithm. Proceedings of 2006 IEEE Congress on Evolutionary Computation. (pp. 954-961). Vancouver, Canada. IEEE. Available [here](#)
© 2006 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This Conference Proceeding is posted at Research Online.
<https://ro.ecu.edu.au/ecuworks/2025>

Fixture-scheduling for the Australian Football League using a Multi-Objective Evolutionary Algorithm

Luigi Barone, *Member, IEEE*, Lyndon While, *Senior Member, IEEE*,
Paul Hughes, and Phil Hingston, *Member, IEEE*

Abstract— AFL football is a team sport that entertains millions and contributes a huge amount of money to the Australian economy. Scheduling games in the AFL is difficult, as a number of different, often conflicting, factors must be considered. In this paper, we propose the use of a multi-objective evolutionary algorithm for determining such a schedule. We detail the technical details needed to apply a multi-objective evolutionary algorithm to this problem and report on experiments that show the effectiveness of this approach. Comparison with actual schedules used in the AFL demonstrates that this approach could make a useful contribution.

I. INTRODUCTION

Australian Rules Football (or AFL football) is a team sport mostly similar to Gaelic Football, and distantly similar to other contact-based team sports like American Football and Rugby. Two teams of 22 players each play against each other in a competition lasting about two hours. The game is renowned for its skill and speed, with elite contestants among the highest paid sports-people in Australia.

The national competition is called the Australian Football League (AFL) [1] and comprises sixteen teams across up the country. Games are played in batches called *rounds*, typically played over the course of a weekend. Due to the physicality of the game, teams are required to play only one game per round, so each round consists of eight games. The regular season lasts 22 rounds, meaning the entire competition consists of 176 games. Note that each team does not play each other team the same number of times. A schedule for all games is known as the *fixture*.

According to the organising authority, AFL football is “Australia’s premier spectator sport attracting more than 14 million people to watch all levels of the game across all communities” [2]. They state that “in 2004, more than 5.9 million people attended premierships season games” (the population of Australia is about 20 million). Their website goes on to discuss the game as an industry: “the game contributes in excess of [A]\$1 billion annually to the Australian economy according to independent research”. A\$780 million was paid for the television rights for the upcoming 2007–2011 period [3], thus making “the AFL the premier sporting competition in Australia”.

Luigi Barone, Lyndon While, and Paul Hughes are with the School of Computer Science & Software Engineering, The University of Western Australia, Crawley 6009, Australia (phone: +61864881944; fax: +61864881089; email: {luigi, lyndon, hughep01}@csse.uwa.edu.au).

Phil Hingston is with the School of Computer and Information Science, Edith Cowan University, Mt Lawley 6020, Australia (phone: +61893706427; email: p.hingston@ecu.edu.au).

Scheduling games for the AFL is difficult, as the organising authority needs to balance a number of different, often conflicting, considerations. Factors like competition fairness, revenue expectations, political considerations, and availability of venues can affect the fixture. As in most sports, home-ground advantage is a significant factor in many games, especially interstate games where travel considerations and time-zone changes affect players, so balancing the number of games a team plays at home and away is important to ensure fairness. Especially considering that each team does not play each other team the same number of times, the task of determining a fixture is open to even more “tuning”. For example, scheduling multiple games between traditional rivals may increase revenue (fans prefer “blockbuster” matches), but this may reduce the fairness of the competition. Trade-offs result — changing the schedule to improve one objective may result in worsening other objectives. This is the realm of multi-objective optimisation, where the task is to find a set of suitably good solutions that vary the trade-offs in the different objectives by differing amounts in order to produce a range of alternative solutions.

Evolutionary-based multi-objective optimisation has been used on a variety of different problems including scheduling [4], but little work has been done in sports fixture scheduling. In this paper, we describe a study which uses a multi-objective evolutionary algorithm to determine a fixture for the AFL, subject to a number of constraints and objectives desired by the organising authority for the game.

The rest of the paper is structured as follows. Section II presents a summary of related work on using optimisation-based approaches for sports fixture scheduling. Section III gives an overview of multi-objective optimisation, including the terminology used in this paper. Section IV describes our multi-objective approach, providing the technical details used to solve this problem. Section V presents results of experiments that demonstrate the effectiveness of our approach, comparing our evolved solutions to the current fixture employed by the AFL organising authority. Finally, Section VI concludes the paper.

II. PREVIOUS APPROACHES TO FIXTURE SCHEDULING

Many approaches have been proposed for solving sports scheduling problems. These problems are especially difficult because each league has its own idiosyncratic requirements, constraints, and preferences. Perhaps for this reason, many previous approaches simply seek any solution that satisfies all the problem constraints — that is, the problem is often

cast as a satisfaction problem rather than an optimisation problem.

For example, a special case is the Sports League Scheduling Problem (*Prob026* of CSPLib [5]), as posed by McAloon *et al.* [6]. In this problem:

- there are T teams, where T is even;
- each team plays each other once, and each team plays one game per round, so there are $T - 1$ rounds;
- there are $\frac{T}{2}$ periods in a round, with one game per period; and
- no team plays more than once in a given period.

Hamiez and Hao [7] solved this problem in linear time when $(T - 1) \bmod 3 \neq 0$, using an exhaustive repair method, improving on earlier results using Tabu search [8] and various earlier approaches including integer linear programming, constraint programming, and randomised complete searches with and without heuristics.

Another special case is the Travelling Tournament Problem, which is concerned with minimising the total distance travelled in a double round robin tournament. Easton *et al.* [9] present a combined integer programming and constraint programming approach to this problem for a group of eight teams. This is much simpler than the AFL scheduling problem, but the authors note that solving even this simplified problem for small numbers of teams is difficult.

A third example is the break minimisation problem (e.g. [10]), which deals with finding a round-robin schedule that minimises the number of consecutive home or away games for the teams.

There has been less success on more “realistic” problems. Carefully formulated methods that rely on the regularity of the problem are all too readily rendered useless when additional constraints or preferences are added. This makes evolutionary algorithms an attractive option, as they can deal with arbitrarily complex objective functions, and there are good techniques available to handle constraints and preferences.

Although evolutionary algorithms have been much used for timetabling and scheduling problems (e.g. [11]), there are only a few examples of evolutionary algorithms being used for sports scheduling. Some examples are: Schönberger *et al.* [12] used a genetic algorithm to schedule the rounds of a table-tennis competition; Yang *et al.* [13] used an evolution strategy to solve sports scheduling problems; Schönberger *et al.* [14] used a memetic algorithm for sports league scheduling; Costa [15] used a hybrid evolutionary Tabu search to schedule hockey leagues; and Yang *et al.* [13] used a genetic algorithm to schedule games for a baseball league. All these studies report good results compared to previously employed methods.

However, we are not aware of any previous work that applies multi-objective evolutionary algorithms to sports scheduling.

III. MULTI-OBJECTIVE OPTIMISATION

Multi-objective optimisation is the task of finding an optimal solution to a problem in which candidate solutions

are judged according to multiple criteria that conflict with each other to some degree. Thus, a good solution can be improved on one criterion only by accepting worse performance in at least one other criterion. The aim in multi-objective optimisation is to generate a set of solutions that compromise the different criteria to varying degrees — the solution to be used in any given situation is selected later according to the particular needs of that situation.

Without loss of generality, consider a multi-objective optimisation problem with a vector of objective functions that maps individuals into fitness space. Given two individuals \vec{a} and \vec{b} , \vec{a} dominates \vec{b} iff \vec{a} is at least as good as \vec{b} in all objectives and better in at least one. \vec{a} is non-dominated with respect to a set X iff there is no individual in X that dominates \vec{a} . X is a non-dominated set iff all individuals from X are mutually non-dominating. The set of corresponding objective vectors is called the non-dominated front.

\vec{a} is Pareto optimal iff \vec{a} is non-dominated with respect to the set of all possible vectors. Such a vector is characterised by the fact that improvement in any one objective necessarily means a worsening in at least one other objective. The Pareto optimal set is the set of all possible Pareto optimal vectors. The goal of multi-objective optimisation is hence to find this Pareto optimal set, although for continuous problems a representative subset suffices.

Since evolutionary algorithms are population based, the partial order imposed on the search space creates a need for an appropriate ranking scheme. Two schemes are commonly employed. Both schemes employ the concept of domination to assign a Pareto rank to individuals — a lower rank implies a superior candidate. In Goldberg’s [16] ranking procedure, non-dominated vectors are assigned a rank of 0 while any dominated vector \vec{a} in the population X is assigned a rank equal to one plus that of the highest-ranked vector from X that dominates \vec{a} . In contrast, Fonseca and Fleming propose a scheme [17] in which a dominated vector \vec{a} in the population X is assigned a rank equal to the number of vectors in X that dominate \vec{a} . It is this Pareto rank, rather than some (weighted) combination of the objectives, that is used as the basis for selection in a multi-objective evolutionary algorithm.

IV. OUR MULTI-OBJECTIVE APPROACH

In this section, we describe the technical details needed to equip a multi-objective evolutionary algorithm to address the problem of fixture determination in the AFL. The multi-objective evolutionary algorithm we use in this work is a variant of NSGA-II [18].

A. Representation

The first step in designing an evolutionary algorithm is determining a representation for candidate solutions suitable for manipulation by genetic operators.

Recall that the AFL competition requires sixteen teams to play 22 games, each team playing one game per round for a total of $8 \times 22 = 176$ games. This means each team does not play each other team the same number of times, and hence the competition is *not* a complete round-robin

tournament [19]. However, the organising authority for the game does impose the following constraint: every team must play each other team in the first fifteen rounds, with the remaining seven rounds the reverse (home and away teams switch) of the first seven rounds. Hence, we need to build only a complete round-robin tournament (where each team plays each other just once) to represent the first fifteen rounds of the competition, and then use the first seven rounds of this round-robin tournament to construct the last seven rounds of the 22 round AFL competition.

A round-robin tournament for an even number of teams n can be constructed using the *polygon construction method* [19]:

- Construct an $n - 1$ sided polygon, and label each vertex and the center point with a team name.
- Draw $\frac{n}{2}$ line segments connecting the vertices or the centre point such that each vertex or the centre point is on only one line segment and no line-segment is a rotation or reflection of another. Each line segment represents the game pairings for a round.
- Rotate the vertex labels $\frac{1}{n-1}$ of a circle by moving each label to the next vertex position to determine the game pairings for the next round.
- Repeat the previous rotation (in the same direction) a further $n - 2$ times to determine all games for all rounds of the round-robin tournament.

The start of this process is shown in Fig. 1 for $n = 8$.

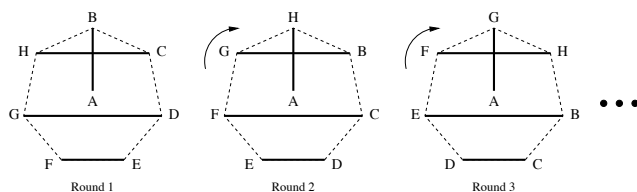


Fig. 1. An example of the polygon construction method for building a round-robin tournament for eight teams.

While this technique can be extended to generate tournaments with optimal home-and-away sequences, experience with previous AFL fixtures indicates this is not necessarily required and is often not employed in favour of improving another objective (e.g. revenue). Indeed, previous AFL fixtures contain many sequences of continuous home (or away) games for certain teams.

We instead use the polygon construction method to build a *round matrix* in which we record the round that each team plays each other. The games for each round are determined from one arrangement of the polygon, and we record the round number in the round matrix. For example, the round matrix generated for the example shown in Fig. 1 is listed in Table I. Note that the matrix always reflects about the main diagonal.

To determine home teams, we use an additional boolean *home team matrix* that indicates which team for each pairing plays at home. This home team matrix is the same size as the round matrix, and also reflects about the main diagonal,

Teams	A	B	C	D	E	F	G	H
A	-	1	7	6	5	4	3	2
B	1	-	4	7	3	6	2	5
C	7	4	-	3	6	2	5	1
D	6	7	3	-	2	5	1	4
E	5	3	6	2	-	1	4	7
F	4	6	2	5	1	-	7	3
G	3	2	5	1	4	7	-	6
H	2	5	1	4	7	3	6	-

TABLE I

FIXTURE FOR THE ROUND-ROBIN TOURNAMENT OF FIG. 1. EACH ENTRY INDICATES THE ROUND IN WHICH THE TWO TEAMS PLAY.

except that each entry below the main diagonal is the negation of the corresponding entry above the main diagonal. As we will see in the Section IV-C, evolutionary selection pressure will drive the algorithm to locate solutions with good home-and-away sequences.

One last constraint of the organising authority must also be handled: a so called *rivalry round* (a round in which each team plays against their traditional rival), must occur twice in the competition. This can be achieved via construction in the polygon method by ensuring that the first round generated by the algorithm is the rivalry round. However, genetic mutation must ensure the pairing of teams in the initial polygon produce the rivalry round. We will see in the next section how this is achieved.

The polygon construction method outlined above generates rounds in the round matrix in a predetermined order. This may induce bias in the search, so instead of using actual team names and round numbers in the construction algorithm, we use logical team names and round numbers, and then convert these logical values to actual values via two maps: a *logical-to-actual team map* for converting team names, and a *logical-to-actual round map* for round numbers. This allows for an unbiased search, offering more exploration of the search space by the evolutionary algorithm.

In summary, a fixture for the AFL is represented by four matrices:

- the round matrix which lists which round in the first fifteen rounds each team plays each other,
- a home team matrix which indicates which team is the home team for any given game pairing,
- a logical-to-actual team map which converts logical team names used in the construction algorithm to actual team names in the AFL, and
- a logical-to-actual round map which converts logical round numbers in the construction algorithm to actual rounds in the AFL fixture.

A complete fixture for the AFL can then be generated from these matrices, observing the constraints about additional rounds (the seven rounds after the round-robin tournament must be the reverse of the first seven rounds), and the rivalry round.

Note that by using this representation, the round matrix need never be modified as all alternative fixtures that con-

stituent a valid robin-round tournament (i.e. fixtures in which teams play precisely once per round) can be obtained by changes to the two logical-to-actual maps and the home team matrix. Indeed, if we did allow modification of the round matrix during evolutionary mutation, modifications are likely to violate this constraint and we would find the search floundering in the large infeasible regions of the search space. For this reason, we allow modification of only three matrices: the home team matrix, the logical-to-actual team map, and the logical-to-actual round map. These three matrices form a candidate solution in our evolving population.

B. Mutation

As we indicated in the previous section, to explore the search space but retain constraint-preserving solutions (fixtures that preserve the round-robin and rivalry round constraints), we require a constraint-preserving mutation operator. As changes to the round matrix are likely to produce invalid solutions and are unnecessary as all alternative fixtures can be generated from changes to the other three matrices, mutation of this matrix is outlawed.

Mutation in our multi-objective algorithm operates on each individual matrix making up a candidate solution (the home team matrix, the logical-to-actual team map, and the logical-to-actual round map). Mutation of the home team matrix is achieved by randomly selecting 0–4 games inclusive and reversing the home team for these games. Mutation of the logical-to-actual round map occurs by simply swapping two co-domain entries in the map. Mutation of the logical-to-actual team map is more complex as it must preserve the rivalry round constraint.

Mutation of the logical-to-actual team map involves two possibilities:

- 1) swap locations of any two teams connected by a line-segment in the initial (first round) polygon of the polygon construction method, or
- 2) swap the two teams connected by any line-segment with two other teams connected by another line segment in the the initial (first round) polygon of the polygon construction method.

Since only the ordering of teams in the initial polygon is modified (and not the actual pairings), the rivalry round constraint is preserved by this mutation scheme.

In order to preserve genetic inheritance (a necessary condition for evolution to proceed), mutation usually involves only small changes to an individual solution. We consider a change to the home team matrix to be a small exploration step, and changes to either of the two logical-to-actual maps as producing much larger evolutionary variations. We hence restrict mutation of the logical-to-actual maps to occur with 5% probability each (both methods of logical-to-actual team map mutation occur with 2.5%, probability), while mutation of the home team matrix is performed on every parent selected for reproduction (note however that $\frac{1}{4}$ of such mutations will yield an unmodified clone).

C. Objectives

Recall that the problem of fixture determination for the AFL is multi-objective, as a number of different, often conflicting, factors must be considered. For consistency, we cast each objective in terms of a minimisation problem.

1) *Home Games*: As indicated previously, playing a game of AFL football at home offers an advantage over playing away. To ensure fairness in the competition, a fixture should try and balance the number of home games each team has. However, continuous sequences of home (or away) games can also affect fairness, as continuous sequences of home (or away) games can affect team morale or “momentum” (a long sequence of away games can result in a run of losses, leading to negative psychological effects on players), or have effects on ticket sales (fans prefer regular home games). Consequently, in terms of equity, not only should the number of home (and away) games be close to balanced for each team, but also the sequences of continuous home (or away) games should also be minimised.

Our first objective captures this equity measure:

$$Equity = \sum_{t \in T} \left(10 \times \left| \frac{R}{2} - H_t \right| + \sum_{i=1}^n L_i^2 \right)$$

where T is the set of all teams, R is the total number of rounds in the competition, H_t is the number of games played at home by team t , n is the number of distinct sequences of the same type (home or away games) for team t , and L_i is the number of games in an individual sequence.

The first part of this measure ($|\frac{R}{2} - H_t|$) captures the requirement that each team play the same number of games at home. The second part ($\sum_{i=1}^n L_i^2$) penalises sequences of continuous games of the same type. We weighted the imbalance in home and away games more highly than the penalty for sequential games as its effect on competition fairness is much more pronounced than sequences of continuous games of the same type. A weighting of 10 was arbitrarily chosen; the equivalent of the penalty for a five game sequence of games of the same type.

2) *Travel*: Australia is a large country (approximately the same size as the continental U.S.), with long distances between the major cities. Interstate travel is hence unavoidable, and coupled with potential changes in time-zones, is generally considered detrimental to team performance. This effect is also believed to increase when travelling for sequential weeks. To produce a fair competition, we need to reduce the effects of such travel. We consider the effects of local travel (within the state) to be insignificant compared to interstate travel.

To capture a measure of interstate travel, we give each state within Australia a location number: Western Australia is 0, South Australia is 1, Victoria is 2, New South Wales is 3, and Queensland is 4. We then use the difference in the location numbers to provide an estimate of the travel required. For example, travel between South Australia and New South Wales scores 2, while travel between Queensland and Western Australia scores 4. We penalise sequential travel

by multiplying travel scores by the number of sequential interstate trips:

$$Travel = \sum_{t \in T} \sum_{i=1}^n \sum_{j=1}^{L_i} T_j \times L_i$$

where T is the set of all teams, n is the number of distinct interstate travel sequences by team t , L_i is the number of games in an individual sequence, and T_j is the travel score for game j .

3) *Expected Revenue*: An important objective in determining an AFL fixture is profit. Profit affects how much the organising authority can spend on promotion and development, and ultimately on salaries of players and officials.

The expected revenue for any game can be estimated via historical data, ground capacity, and the expected television audience. Cast as a minimisation problem, we negate the sum of the expected revenues for all games in the fixture to give us a measure of expected revenue:

$$Revenue = - \sum_{i=1}^G rev(t_1, t_2)$$

where G is the total number of games in the competition, and $rev(t_1, t_2)$ is the expected revenue for a game involving teams t_1 and t_2 , team t_1 playing at home.

4) *Venue Distribution*: The AFL competition consists of sixteen teams in five states: ten in Victoria, two in South Australia, two in Western Australia, one in New South Wales, and one in Queensland. Distributing the games across the country is important for a number of reasons, including political reasons (e.g. contractual obligations), venue availability, and maintenance concerns (venues can be over-used, degrading the quality of the playing surface). Since most teams based in the same state share the same venue in order to reduce overheads, venue availability is an importance consideration. Note that smaller venues typically bring in less revenue than the larger venues.

Indeed, experience shows that the organising authority of the AFL has a preferred number of games for each state per round: five in Victoria, one in South Australia, and one in Western Australia. We capture this concept by recording the number of games played in each state in each round, and noting the difference of this number from the preferred number:

$$Distribution = \sum_{i=1}^R \sum_{s \in S} |P_s - A_s|$$

where R is the total number of rounds, S is the set of states with more than one team, P_s is the preferred number of games to be played in state s in one round, and A_s is the actual number of games played in state s in round i .

We do not consider games played in New South Wales and Queensland, as these states host only one team and hence have no need to share home venues with other teams. Indeed, the 2006 AFL fixture has no discernible pattern for the games in these two states.

D. Selection

Determination of which candidate solutions survive and reproduce in our multi-objective evolutionary algorithm is based on Pareto rank. We use the Pareto ranking scheme proposed by Fonseca and Fleming [17], but we modify the final rank of each solution to bias the search to preferred regions of the search space. We explain the motivation and implementation of this scheme below.

Recall that by using the polygon construction method and a constraint-preserving mutation operator, we can ensure that all individual solutions examined by the evolutionary algorithm are valid, meeting all the constraints imposed by the organising authority. However, note that allowing the mutation operator the freedom to modify the home team matrix opens up large regions of the search space that involve fixtures in which every team does not play the same number of home and away games. Indeed, the size of the search space corresponding to changes in just the home team matrix is extremely vast — much more so than the search space corresponding to changes to just the two logical-to-actual maps.

Note that an unequal number of home and away games does not render a solution infeasible (the solution does not violate any constraints imposed by the AFL organising authority), just that it is not preferred as it is unlikely to be adopted because of the inequity it represents. In essence, home-and-away game balance represents some form of “soft constraint” — the organising authority would like all teams to play the same number of home and away games, but are willing to consider “limited” violations of the constraint for improvements in other objectives.

From an optimisation algorithm stand-point, what is sought is some way of controlling the search — concentrating the search in the preferred regions of the search space, but still allowing some search in the non-preferred regions as they may contain solutions of interest. While the equity objective outlined in the previous section achieves this to some extent by penalising solutions that contain an unequal number of home and away games, non-preferred solutions will still emerge as they can achieve a good Pareto rank by performing well in another objective. Over time, evolutionary selection pressure will hopefully drive the search to find solutions that minimise this mismatch and also perform well in the other objectives (thus removing these non-preferred solutions), but many fitness evaluations (or generations) may be required. Some other means of “fast-tracking” the search away from these regions is desired.

To achieve this, instead of simply performing selection on Pareto rank, we first modify the rank of each solution by penalising solutions that contain a relatively large inequity in the number of home and away games each team plays, and then base selection on these modified ranks. A random number is also added to each solution’s rank to avoid ties. The modified rank used in the selection process is hence

calculated as:

$$Rank = Pareto\ rank + f \times \sum_{t \in T} (H_t - H_{best})^2 + Rand()$$

where *Pareto rank* is the standard Pareto rank of the solution, *Rand()* is a randomly sampled uniform variable from the range $[0, 1]$, *f* is a scaling factor that determines the relative importance of the inequity penalty, *T* is the set of all teams, H_t is the number of games played at home by team *t*, and H_{best} is the closest number to the optimal number of home games ($\frac{R}{2}$, where *R* is the number of rounds in the competition) for any solution in the population.

We see from this formula that the penalty function has a small effect at the start of the run (where high ranking solutions still have a good chance of being selected by the evolutionary algorithm as local optimisation has yet to increase the number of low ranked solutions), thus allowing exploration of the non-preferred regions of the search space. When progress towards the Pareto optimal front starts to slow (and local optimisation increases the number of lowered ranked solutions), the penalty function will increase evolutionary selection pressure towards solutions that contain the same number of home and away games. The net effect of this is a bias in the search, limiting exploration of the vast regions of the search space that contains many non-preferred solutions, concentrating on regions that are likely to contain solutions of interest. In essence, this approach allows us to provide some form of ordering over the objectives — the equity objective is, to some extent, more important than the other three objectives.

The scaling factor *f* allows us to control the effects of the inequity penalty on rank. We will see in the next section that this parameter has significant effects on the percentage of solutions in the final population that contain an unequal number of home and away games.

V. EXPERIMENTAL RESULTS

The multi-objective evolutionary algorithm we use in this work is a hybrid of NSGA-II [18]. Selection is determined using the ranking scheme detailed in Section IV-D. We use an elitism rate of 50%, thus meaning we preserve the best 50% of the population from one generation to generation. Initial experiments demonstrate that a population of 800 yields good results in a few hours.

A. Adjustment Factor Determination

Our first experiment aims at determining the value of the scaling factor *f* in the modified ranking scheme employed by our multi-objective algorithm.

Fig. 2 plots the percentage of solutions that contain an unequal number of home and away games for different values of the scaling factor. Each run of the multi-objective evolutionary algorithm lasted 10,000 generations and used a population of 800 candidate solutions. Each point is the average of ten runs of the evolutionary algorithm.

As evidenced from Fig. 2, without any rank adjustment, almost all solutions in the final population contain an unequal

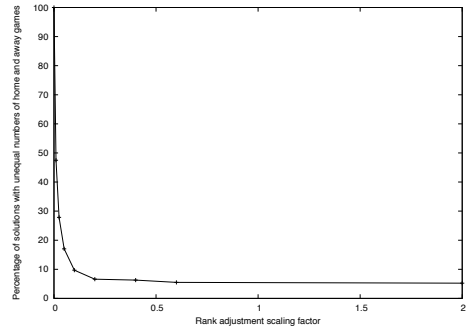


Fig. 2. Percentage of solutions that contain an unequal number of home and away games for different values of the adjustment factor used in the ranking scheme employed by our multi-objective evolutionary algorithm.

number of home and away games. As we increase the effect of rank adjustment, the percentage of non-preferred solutions in the final population falls away quickly. Indeed, with a scaling factor of 0.25, over 94% of resultant solutions contain an equal number of home and away games. We settle on a scaling factor of 0.25 for our future experiments.

B. Pareto Front Evolution

Fig. 3 plots the different perspective views of the Pareto front at different stages (generations) during a single run of our multi-objective evolutionary algorithm. Only solutions with an equal number of home and away games are plotted, and for each perspective view, only non-dominated solutions with respect to those two objectives are shown. We use a population of 800 candidate solutions and a rank adjustment scaling factor of 0.25 in our multi-objective evolutionary algorithm.

Fig. 3 shows that our multi-objective evolutionary algorithm is able to simultaneously optimise all objectives. Note the progression of the fronts towards the idealised minimum (the origin), and note there is only one crossover in fronts from different generations (the front after 2000 generations is “behind” the front after 1,000 generations in Fig. 3(e)). This confirms that our algorithm is able to maintain a good rate of progression during the entire run.

The crossover of fronts for Fig. 3(e) suggests a loss of an equal home-and-away game solution from the 1,000 generation population that would have been non-dominated in the 2000 generation population. This is probably the result of random selection — occurring when the algorithm is forced to choose between equally ranked solutions (recall that we add a random number to the Pareto rank of each solution, effectively resolving ties in adjusted ranks randomly).

Fig. 3 also shows that the evolutionary algorithm is able to generate a good range of different solutions, each trading-off the different objectives by varying amounts. During the course of the run we see good coverage across each objective, highlighting the evolutionary algorithm’s ability to explore different parts of the search space. The general narrowing of the Pareto front towards the end of run highlights that the algorithm is able to locate a few solutions good in

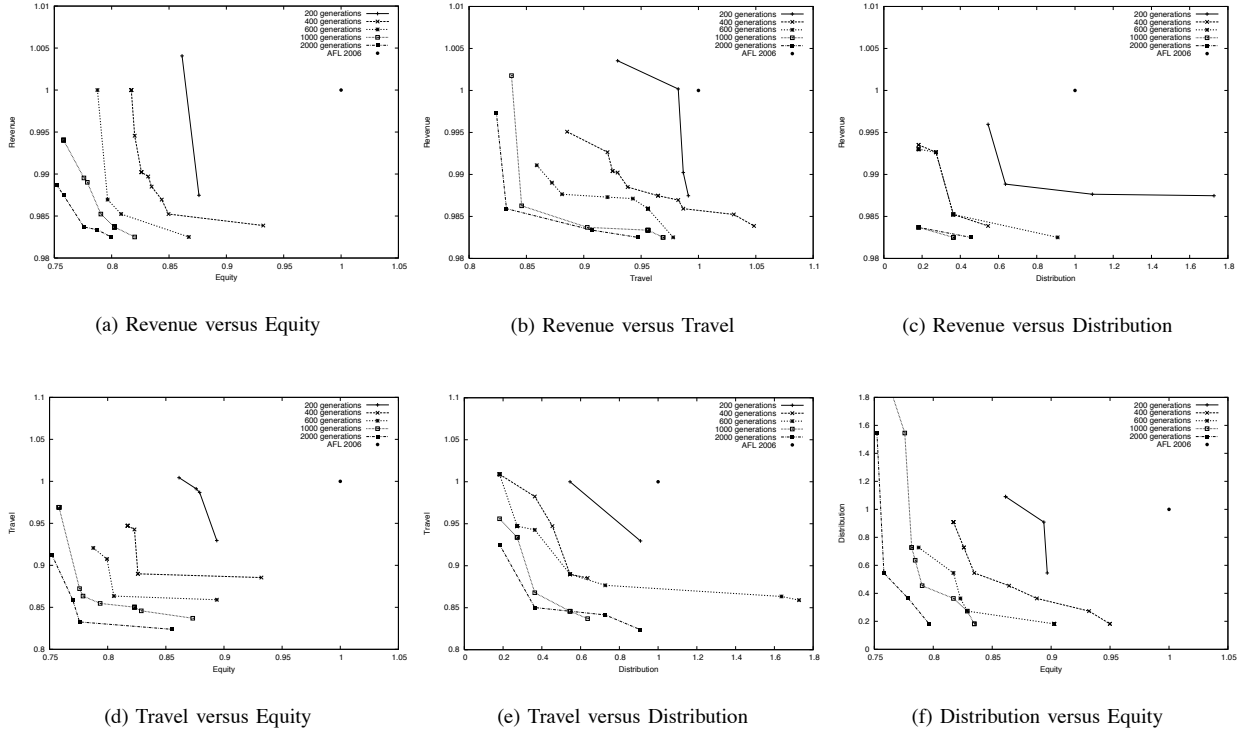


Fig. 3. Pareto front evolution for one run of our multi-objective evolutionary algorithm. Objective values are normalised by the values for the AFL 2005 fixture.

several objectives, suggesting some interdependence between the objectives.

C. Hypervolume

The *hypervolume* metric for non-dominated front comparison [20], [21] measures the ratio of the hypervolume dominated by a front to the hypervolume dominated by the idealised minimum. It provides a numerical measure that rewards both closeness to the Pareto optimal front and the extent of the obtained non-dominated front. Importantly, the hypervolume metric is more robust than other numerical metrics [22], [23], [24].

Fig. 4 plots the hypervolume of the Pareto front at different stages (generations) during a single run of our multi-objective evolutionary algorithm. Fig. 4 also plots the hypervolume contributed by just solutions with an equal number of home and away games.

Fig. 4 confirms our earlier observation about a good rate of progression for our evolutionary algorithm — we observe a general increase in the hypervolume of the Pareto front over time. Fig. 4 also shows the effects of the rank adjustment scheme used in this work. We observe that initially, all of the Pareto front consists of the non-preferred solutions that have an unequal number of home and away games. Over time, we see an increase in the proportion of the hypervolume contributed by preferred solutions, suggesting an emergence of these solutions in the population. This gives us the effect we seek — some exploration of non-preferred solutions early

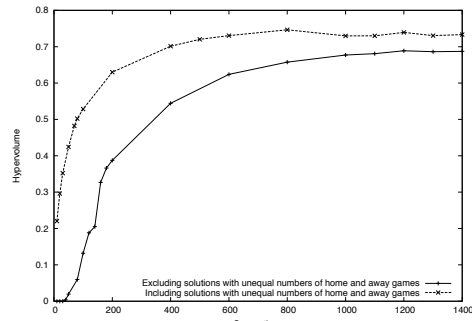


Fig. 4. Progression of the hypervolume of the Pareto front for one run of our multi-objective evolutionary algorithm.

in the run, with a general convergence to preferred solutions towards the end of the run.

D. Comparison with the Current AFL Fixture

Also marked on each plot in Fig. 3 is the current fixture employed by the AFL organising authority for the 2006 season. Examination of this figure shows that our evolutionary algorithm is quickly able to produce solutions that dominate this schedule.

Table II compares selected fixtures taken from ten runs of our evolutionary algorithm with the 2006 AFL fixture.

We see from Table II that our multi-objective evolutionary algorithm is able to produce a good range of solutions that

Solution	Objective			
	Equity	Distribution	Travel	Revenue
AFL 2006	1.00	1.00	1.00	1.000
Best revenue	0.82	0.64	1.05	0.977
Second best revenue	0.81	0.18	0.97	0.978
Median revenue	0.75	1.27	0.89	0.986
Best travel	0.81	1.45	0.82	0.989
Good travel	0.77	1.64	0.89	0.984
Best equity	0.71	1.64	1.00	0.992
Good equity	0.73	2.00	0.94	0.984
Best distribution	0.81	0.18	0.97	0.978
Median distribution	0.77	0.82	0.89	0.985

TABLE II
SELECTED FIXTURES FROM TEN RUNS OF OUR MULTI-OBJECTIVE
EVOLUTIONARY ALGORITHM.

trade-off the different objectives by varying amounts. Comparison of the different fixtures shows that the evolutionary algorithm is able to “tweak” generalist solutions in order to optimise performance in any single objective.

The good revenue optimising fixtures reported in Table II ensure that the large earning games (typically those played in Victoria, which has a much larger venue than any other state) appear in the first seven rounds, thus ensuring they appear twice in the final fixture (recall, the first seven rounds are repeated to make the 22 round fixture). The good travel solution has less of these large earning games within the first seven rounds, sacrificing revenue, but minimising the number of interstate trips required by non-Victorian based teams.

The optimisation of the travel objective has a significant negative impact on the distribution objective, with the good travel solution of Table II representing a fixture that contains nil or two games played in both Western Australia and South Australia in ten rounds of the competition. This violates another “soft constraint” of the organising authority — the desire that precisely one game be played in both of these states each round. For this reason, the organising authority is unlikely to employ this fixture for actual use. However, the second best revenue solution (which is also the best distribution solution) of Table II strictly dominates the current AFL schedule, improving performance in every objective: it could potentially be used by the AFL organising authority.

VI. CONCLUSIONS

In this paper, we have presented a multi-objective evolutionary algorithm for fixture determination for the sport of AFL football. Like many team sports that involve teams spread over significant distances, fixture designers for the AFL face the difficult problem of balancing a number of different, often conflicting, factors like competition fairness, amount of travel, availability and distribution of games, and of course revenue.

Our multi-objective approach to this problem produces a range of different fixtures, each varying the trade-offs in the objectives by differing amounts. This provides the organising authority the ability to explore different “what if” options,

allowing them to choose the option that best suits their requirements. Our experiments show that this multi-objective approach is able to evolve solutions that strictly dominate the existing fixture, promising better returns in every measure of success.

REFERENCES

- [1] “Official website of the Australian Football League,” January 2006, URL: <http://afl.com.au/>.
- [2] “The AFL explained,” January 2006, URL: <http://afl.com.au/default.asp?pg=aflinfosheets&spg=display&articleid=240727>.
- [3] “Seven and ten win all rights,” January 2006, URL: <http://www.abc.net.au/sport/content/200601/s1542491.htm>.
- [4] C. A. Coello Coello and G. B. Lamont, Eds., *Applications of Multi-objective Evolutionary Algorithms*. World Scientific Publishing, 2004.
- [5] B. Huich, I. Miguel, I. P. Gent, and T. Walsh, “CSPLib: A problem library for constraints,” January 2006, URL: <http://csplib.org>.
- [6] K. McAloon, C. Tretkoff, and G. Wetzel, “Sports league scheduling,” in *Third ILOG Optimization Suite International User’s Conference*, 1997.
- [7] J.-P. Hamiez and J.-K. Hao, “A linear time algorithm to solve the sports league scheduling problem (prob026 of CSPLib),” *Discrete Applied Mathematics*, vol. 143, pp. 252–265, 2004.
- [8] —, “Solving the sports league scheduling problem with tabu search,” *LNAI*, vol. 2148, pp. 24–36, 2001.
- [9] K. Easton, G. Nemhauser, and M. Trick, “Solving the travelling tournament problem: A combined integer programming and constraint programming approach,” *LNCS*, vol. 2740, pp. 100–109, 2003.
- [10] R. Miyashiro and T. Matsui, “Round-robin tournaments with a small number of breaks,” Department of Mathematical Informatics, The University of Tokyo, Mathematical Engineering Technical Reports METR 2003-29, 2003.
- [11] E. Burke and P. Ross, Eds., *The Practice and Theory of Automated Timetabling*. Springer Verlag, 1996.
- [12] J. Schönberger and H. K. Dirk C. Mattfeld, “Automated timetable generation for rounds of a table-tennis league,” in *CEC 2000*, pp. 277–284.
- [13] J. T. Yang, H.-D. Huang, and J.-T. Horng, “Devising a cost effective baseball scheduling by evolutionary algorithms,” pp. 1660–1665.
- [14] D. C. M. J. Schönberger and H. Kopfer, “Memetic algorithm timetabling for non-commercial sport leagues,” *European Journal of Operational Research*, vol. 153, no. 1, pp. 102–116, 2004.
- [15] D. Costa, “An evolutionary tabu search algorithm and the NHL scheduling problem,” Department of Mathematics, Swiss Federal Institute of Technology, Tech. Rep. ORWP 92/11, 1992.
- [16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, 1989.
- [17] C. M. Fonseca and P. J. Fleming, “Genetic algorithms for multiobjective optimization: formulation, discussion and generalization,” in *5th International Conference on Genetic Algorithms*, 1993, pp. 416–423.
- [18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.
- [19] J. Dinitz, E. Lamken, and W. Wallis, “Scheduling a tournament,” in *Handbook of Combinatorial Designs*, J. Dinitz and C. Colbourn, Eds. CRC Press, 1995, pp. 578–584.
- [20] E. Zitzler, “Evolutionary algorithms for multiobjective optimization: methods and applications,” Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.
- [21] L. While, P. Hingston, L. Barone, and S. Huband, “A faster algorithm for calculating hypervolume,” *IEEE TEC*, vol. 10, no. 1, pp. 29–38, 2006.
- [22] J. Knowles and D. Corne, “On metrics for comparing nondominated sets,” in *CEC 2002*, pp. 711–716.
- [23] K. H. Ang, G. Chong, and Y. Li, “Preliminary statement on the current progress of multi-objective evolutionary algorithm performance measurement,” in *CEC 2002*, pp. 831–836.
- [24] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE TEC*, vol. 7, no. 2, pp. 117–132, 2003.