Referee Assignment in Sports Tournaments

Celso C. Ribeiro*
Alexandre R. Duarte
Sebastián Urrutia

EURO XXI

Reykjavik, Iceland, July, 2006
Summary

• Motivation
• Referee assignment
• Problem statement
• Solution approach
  – Greedy constructive heuristic
  – Local search and neighborhoods
  – ILS-based scheme
  – Repair heuristic
  – Improvement heuristic
• Numerical results
• Conclusions and future work
Motivation

- Optimization in sports is a field of increasing interest
  - Traveling tournament problem
  - Playoff elimination
  - Tournament scheduling
  - Referee assignment

- Regional amateur leagues in the US (baseball, basketball, soccer): hundreds of games every weekend in different divisions
Motivation

- In a single league in California there might be up to 500 soccer games in a weekend, to be refereed by hundreds of certified referees.

- MOSA (Monmouth & Ocean Counties Soccer Association) League (NJ): boys & girls, ages 8-18, six divisions per age/gender group, six teams per division: 396 games every Sunday (US$ 40 per ref; US$ 20 per linesman, two linesmen).

- Problem: assign referees to games.

- Referee assignment involves many constraints and multiple objectives.
Referee assignment

• **Possible constraints:**
  – Different number of referees may be necessary for each game
  – Games require referees with different levels of certification: higher division games require referees with higher skills
  – A referee cannot be assigned to a game where he/she is a player
  – Timetabling conflicts and traveling times
Referee assignment

• **Possible constraints (cont.):**
  
  – Referee groups: cliques of referees that request to be assigned to the same games (relatives, car pools)
    
    • Hard links
    
    • Soft links
  
  – Number of games a referee is willing to referee
  
  – Traveling constraints
  
  – Referees that can officiate games only at a certain location or period of the day
Referee assignment

- Possible objectives:
  - Difference between the target number of games the referee is willing to referee and the number of games he/she is assigned to
  - Traveling time between consecutive games
  - Number of inter-facility travels
  - Waiting time between consecutive games
  - Number of games assigned outside his/her preferred time-slots or facilities
  - Number of violated soft links
Problem statement

• Games are already scheduled (facility – time slot)
• Each game has a number of refereeing positions to be assigned to referees
• Each refereeing position to be filled by a referee is called a refereeing slot
• $S = \{s_1, s_2, \ldots, s_n\}$: refereeing slots to be filled by referees
• $R = \{r_1, r_2, \ldots, r_m\}$: referees
Problem statement

- \( p_i \): skill level of referee \( r_i \)
- \( q_j \): minimum skill level a referee must have to be assigned to refereeing slot \( s_j \)
- \( M_i \): maximum number of games referee \( r_i \) can officiate
- \( T_i \): target number of games referee \( r_i \) is willing to officiate
- Each referee may choose a set of time slots where he/she is not available to officiate
Problem statement

• **Problem:** assign a referee to each refereeing slot

• **Constraints:**
  – Referees officiate in a single facility on the same day
  – Minimum skill level requirements
  – Maximum number of games
  – Timetabling conflicts and availability

• **Objective:** minimize the sum over all referees of the absolute value of the difference between the target and the actual number of games assigned to each referee (0-1 integer linear programming model)
Solution approach

- Three-phase heuristic approach
  1. Greedy constructive heuristic
  2. ILS-based repair heuristic to make the initial solution feasible (if necessary)
  3. ILS-based procedure to improve a feasible solution
Solution approach

**Algorithm RefereeAssignmentHeuristic** (MaxIter)

1. \( S^* \leftarrow \text{BuildGreedyRandomizedSolution}() \);
2. **If not** isFeasible \((S^*)\) **then**
3. \( S^* \leftarrow \text{RepairHeuristic} (S^*, \text{MaxIter}); \)
4. **If** isFeasible \((S^*)\) **then**
5. \( S^* \leftarrow \text{ImprovementHeuristic} (S^*); \)
6. **Else** “infeasible”
7. **Return** \( S^* \)
Greedy constructive heuristic

• Assign refereeing slots to referees that are also players (hard facility constraint)
• While there are unassigned refereeing slots and unassigned referees do:
  – Select the highest skill level (hSL) among all unassigned referees
  – Greedily select the facility with unassigned slots with the strongest need for a referee with skill level hSL
  – Assign a referee with skill level hSL to refereeing slots in this facility
• Complete the solution with infeasible assignments
Local search and neighborhoods

• Solutions built by a construction algorithm are not necessarily optimal.

• Local search algorithm successively replaces the current solution by a better one in a neighborhood of the first, terminating at a local optimum.

• First improving strategy: the current solution is replaced by the first neighbor whose cost function value improves that of the current solution.
Local search and neighborhoods

- **Swap moves**: referees assigned to two refereeing slots are swapped (number of games assigned to each referee does not change)
- **Exchange moves**: referee assigned to a refereeing slot is replaced by another referee (number of games assigned to each referee either increases or decreases by one unit)
- **Only moves involving referees that officiate at the same facility (or do not officiate at all) are allowed**
ILS-based scheme

- Both the repair and the improvement heuristics use similar ILS (iterated local search) schemes:
  - Apply first improving local search (first exchange moves, next swap moves) to the initial solution
  - For a given number of iterations, apply a perturbation to the current solution, followed by local search considering only the facilities involved in the perturbation, and accept the new solution if it is better than the current solution
**Algorithm ILS_Scheme (S, MaxIter)**

1. **For** each facility f **do**
   2. \( S \leftarrow \text{LocalSearch}(f, S) \);
3. **For** \( i = 1 \ldots \text{MaxIter} \) **do**
4. \( S^{\text{current}}, f_1, f_2 \leftarrow \text{Perturbation}(S) \);
5. \( S^{\text{current}} \leftarrow \text{LocalSearch}(f_1, S^{\text{current}}) \);
6. \( S^{\text{current}} \leftarrow \text{LocalSearch}(f_2, S^{\text{current}}) \);
7. \( S \leftarrow \text{AcceptanceCriterion}(S, S^{\text{current}}) \);
8. \( S^* \leftarrow \text{UpdateGlobalBestSolution}(S, S^*) \);
9. **Return** \( S^* \)
Repair heuristic

• Referees do not travel in the solutions built by the greedy constructive heuristic

• Possibly violated constraints:
  – timetabling conflicts
  – referee availability
  – skill levels
  – maximum number of games

• Approach: minimize the number of violations (no violations in feasible solutions)
Repair heuristic

• Perturbation
  – Select a facility $f_k$ with infeasible assignments
  – Select the maximum minimum skill level over all refereeing slots in this facility assigned to referees with at least one violation
  – Search for a referee $r_i$ that officiates at another facility (or does not officiate at all) whose skill level is at least as large as the above
Repair heuristic

- **Perturbation (cont.)**
  - Randomly select referees that officiate at the same facility to be assigned to the refereeing slots currently assigned to $r_i$
  - Finally, assign referee $r_i$ to a refereeing slot at facility $f_k$ which is currently assigned to a referee with at least one violation
Improvement heuristic

- Minimize the sum over all referees of the difference between the target and the actual number of games assigned to each of them
- Only exchange moves and perturbations that maintain feasibility are considered
- Swap moves are not applied, since they cannot reduce the value of the objective function
Improvement heuristic

- **Perturbation**
  - Select two referees officiating at different facilities, such that the swap of all their assignments is feasible
  - Look ahead: check if there are other games that could be officiated by these referees after the swap
    - Only refereeing slots assigned to referees that are officiating too many games (more than their target) are considered (exchange moves involving other referees cannot reduce the objective function)
  - The first pair of referees whose swap can reduce the objective function is selected and all their assignments are swapped
Numerical results

- Randomly generated instances following patterns similar to real-life applications
- Instances with up to 500 games and 1,000 referees
  - Different number of facilities
  - Different patterns of the target number of games
- Five different instances for each configuration
- MaxIter = 1,000
Numerical results

- For each test instance, average time and average objective value over ten runs
- Codes implemented in C
- Results obtained on a 2.0 GHz Pentium IV processor with 256 Mbytes of RAM
- Initial solutions:
  - greedy constructive heuristic
  - random assignments (to test the repair heuristic)
## Numerical results

<table>
<thead>
<tr>
<th>Instances</th>
<th>Constructive heuristic</th>
<th>Repair heuristic</th>
<th>Improvement heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method</td>
<td>$T_1$ (s)</td>
<td>$Obj_1$</td>
</tr>
<tr>
<td>I1</td>
<td>greedy</td>
<td>0.02</td>
<td>1277.80</td>
</tr>
<tr>
<td>I2</td>
<td>greedy</td>
<td>0.02</td>
<td>1359.60</td>
</tr>
<tr>
<td>I3</td>
<td>greedy</td>
<td>0.02</td>
<td>1275.80</td>
</tr>
<tr>
<td>I4</td>
<td>greedy</td>
<td>0.03</td>
<td>—</td>
</tr>
<tr>
<td>I5</td>
<td>greedy</td>
<td>0.02</td>
<td>1326.00</td>
</tr>
<tr>
<td>I1</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I2</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I3</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I4</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I5</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 1: Instances with 500 games, 750 referees e 65 facilities
## Numerical results

<table>
<thead>
<tr>
<th>Instances</th>
<th>Constructive heuristic</th>
<th>Repair heuristic</th>
<th>Improvement heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method</td>
<td>$T_1$ (s)</td>
<td>Obj$_1$</td>
</tr>
<tr>
<td>I1</td>
<td>greedy</td>
<td>0.03</td>
<td>—</td>
</tr>
<tr>
<td>I2</td>
<td>greedy</td>
<td>0.03</td>
<td>—</td>
</tr>
<tr>
<td>I3</td>
<td>greedy</td>
<td>0.03</td>
<td>—</td>
</tr>
<tr>
<td>I4</td>
<td>greedy</td>
<td>0.03</td>
<td>—</td>
</tr>
<tr>
<td>I5</td>
<td>greedy</td>
<td>0.03</td>
<td>—</td>
</tr>
<tr>
<td>I1</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I2</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I3</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I4</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I5</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 2: Instances with 500 games, 750 referees and 85 facilities
# Numerical results

<table>
<thead>
<tr>
<th>Instances</th>
<th>Constructive heuristic</th>
<th>Repair heuristic</th>
<th>Improvement heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method</td>
<td>$T_1$ (s)</td>
<td>Obj$_1$</td>
</tr>
<tr>
<td>I1</td>
<td>greedy</td>
<td>0.03</td>
<td>1582.00</td>
</tr>
<tr>
<td>I2</td>
<td>greedy</td>
<td>0.03</td>
<td>1627.40</td>
</tr>
<tr>
<td>I3</td>
<td>greedy</td>
<td>0.02</td>
<td>1539.40</td>
</tr>
<tr>
<td>I4</td>
<td>greedy</td>
<td>0.03</td>
<td>1663.20</td>
</tr>
<tr>
<td>I5</td>
<td>greedy</td>
<td>0.02</td>
<td>1623.80</td>
</tr>
<tr>
<td>I1</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I2</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I3</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I4</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>I5</td>
<td>random</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 3: Instances with 500 games, 875 referees and 65 facilities
Conclusions and future work

• Formulation of a new optimization problem in sports
• Effective heuristics (construction, repair, improvement)
• Extensions:
  – Additional constraints
  – Other objectives
  – Multi-objective approach
  – Application to real-life instances