Vol. 37, No. 1 (2027) | DOI: 10.37190/ord/236168

OPEN ACCESS

Operations Research and Decisions

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

Referee assignment problem with multiple leagues in non-professional football

Tankut Atan*[1] Burak Çavdaroğlu[2] Zühal Özcan[1]

[1]*Department of Industrial Engineering, Bahçeşehir University, Beşiktaş, İstanbul, Türkiye, 34353*
[2]*Newcastle Business School, Northumbria University, Newcastle upon Tyne, UK, NE1 8ST*
**Corresponding author, email address: sabritankut.atan@bau.edu.tr*

Abstract

This study introduces the Multi-League Referee Assignment Problem (MRAP), a practically motivated optimization problem arising in the scheduling of referees for non-professional football. MRAP accounts for multiple concurrent leagues with distinct scheduling constraints, qualification requirements, and fairness considerations. We formulate MRAP as an integer programming model and assess its performance through extensive numerical experiments on realistic synthetic data. The monolithic model becomes computationally infeasible for large instances. Therefore, we propose two heuristic approaches that decompose the problem temporally by weeks or structurally by leagues. These heuristics quickly produce high-quality solutions when referee availability is sufficient. Notably, shorter batches of weeks in temporal decomposition perform better, as longer batches create restrictive commitments. Sensitivity analyses further demonstrate the practical value of optimization-based decision support.

Keywords: *OR in sports, round-robin tournaments, integer programming, heuristic methods, decision support*

1. Introduction

Assigning referees to sports competitions is a complex and recurring operational task typically managed by organizing committees. While this task arises in all levels of competition, it is particularly well-suited to benefit from quantitative decision support in non-professional sports. In professional leagues, referee assignments are often subject to intense public scrutiny, especially following controversial or subpar officiating [24]. As a result, assignments are typically made incrementally (week by week or for short horizons) based on referees' recent performances [34]. The high-stakes environment and political sensitivities in professional sports can hinder the effective use of mathematical optimization tools. In contrast, non-professional sports offer a more structured and less contentious setting. Assignments are often determined in advance, based on a predefined pool of referees and their qualifications [11], with little to

Received 2 January 2026, accepted 8 September 2026, published online 8 September 2026
ISSN 2391-6060 (Online)/© 2026 Authors

no real-time public feedback. This makes non-professional contexts more amenable to optimization-based approaches. Moreover, the sheer volume of games and referees involved in these settings further amplifies the need for systematic, scalable assignment tools.

In this article, we will focus on assigning referees in semi-professional and amateur football (i.e., non-professional football) at least when giving examples or using related terminology. The non-professional setting is quite similar in many other mass sports as well. Non-professional football involves several leagues at different age groups with many groups for the same age. Group matches are usually organized either as a round-robin tournament with or without a playoff stage or as a knockout tournament deciding the winner. In any case, for a chosen planning horizon there may be hundreds or even thousands of games for which referees have to be appointed. For example, in the English football league system after the 2021 FA restructuring [27], Level 6 (National League North/South) and below are considered predominantly semi-professional while Level 9 (Step 5 of the National League System) and downward are typically considered amateur. The total number of matches played in a typical week between Levels 6 and 11 of the English football pyramid is more than 800. As another example, only in the city of İstanbul, there are between 500 and 700 non-professional games over a weekend. The referees in these competitions typically hold full-time employment in other professions, with officiating serving as a secondary activity for which they receive modest compensation.

Referees are categorized into distinct qualification levels based on their capabilities and credentials. These classifications reflect multiple factors, including performance evaluations from higher-tier leagues, completion of mandatory educational programs, and successful completion of physical fitness assessments. Consequently, not all referees possess the qualifications required to officiate matches in upper-level leagues. From a planning perspective, it may also be desirable to ensure that referees officiate a balanced distribution of games across different leagues. This serves dual objectives: providing referees with diverse experience to support their professional development, and promoting fairness in both assignment opportunities and compensation. However, these experiences and equity considerations introduce a fundamental scheduling tradeoff. When a referee is assigned to officiate a match in one league, they become unavailable for concurrent matches in other leagues, thereby constraining the overall assignment flexibility within the multi-league system.

This research deals with the Referee Assignment Problem (RAP) in a – differently from the existing literature – *multi-league* setting which we will be referring to as the Multi-League Referee Assignment Problem (MRAP). We will first give a literature review in Section 2 and then make a formal definition of the problem in Section 3. An integer mathematical formulation will be provided followed by descriptions of heuristic approaches (Section 4) we developed for solving the problem. After reporting our numerical results using realistic synthetic data in Section 5, the article will be concluded in Section 6. This research serves as a proof-of-concept of our methodologies for this problem class tested with diverse scenarios such as referee unavailability through extensive numerical experiments; Section 6 also suggests a roadmap for a potential implementation.

2. Literature review

The RAP deals with optimally assigning the referees of a single tournament subject to some constraints. The objective is minimizing 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 [11]. There is also a version similar to the Traveling Tournament Problem, namely the Traveling Umpire Problem where the total travel times of the referees are minimized.

Early work on the RAP by Evans et al. [15] and Evans [14] focused on baseball tournaments and introduced considerations like referee travel times, limits on the number of games refereed for the same team, and balancing referee assignments across teams. Scarelli and Narula [25] proposed applying multi-criteria decision-making techniques to referee assignments, using expert opinions. Similarly, Lafuente [18] suggested solving assignment problems by using weekly referee performance scores provided by experts. Both approaches resemble weekly committee meetings held to make referee assignment decisions. Dinitz and Stinson [9] employed Room Squares to make referee assignments, but this method is unsuitable for incorporating additional custom constraints. In contrast, Duarte et al. [11] and a follow-up work [10] developed a general RAP formulation that accounted for practical constraints such as ensuring referees met the minimum performance rating required for games and avoiding assignments where referees were unavailable. They also introduced a heuristic algorithm based on iterated local search to solve the problem and demonstrated its NP-hard nature. While the RAP formulation by Duarte et al. [11] considered assignments as a set of games, this study acknowledges that games belong to different leagues. Accordingly, we model the problem to reflect league structures, allowing heuristic approaches to prioritize assignments based on league levels and group games by their respective leagues.

Assignment constraints observed in practice vary depending on the tournament's location and type. For example, studies have explored referee assignments in Turkey [34], Chile [1, 21], Italy [22], Portugal [21], and the Netherlands [32]. Farmer et al. [16] devised a methodology for assigning tennis referees at the US Open, while Durán et al. [12] addressed referee assignments in the Argentine basketball league by solving an integer linear model aimed at minimizing referee travel times. Hawkey [17] explored using machine learning to assign referees for Major League Soccer games based on positional, event, and technical performance data. Referees often handle multiple games in one trip, leading to the Traveling Umpire Problem, which minimizes referee travel distances while considering constraints like limiting repeat assignments to the same team and ensuring each referee officiates for every team at least once. The Traveling Umpire Problem is particularly relevant in baseball [29–31], where referees are assigned as teams. Solution techniques for this complex problem include simulated annealing [31], and exact methods [28, 33]. Recent work by Chandrasekharan et al. [6] introduces a matheuristic approach to the problem. Some researchers have combined the RAP with tournament scheduling to solve integrated problems. Bender and Westphal [4] merged the Traveling Tournament Problem with the Traveling Umpire Problem. Using Turkey's first-division football league as a case study, Atan and Hüseyinoğlu [2] proposed an integer model for an integrated referee assignment and tournament scheduling problem, complemented by a genetic algorithm to solve it.

Although not in the context of referee assignments, some academic work dealt with multiple leagues for simultaneously scheduling the leagues' games. Moody et al. [23] provide a heuristic algorithm for creating timetables for several youth leagues where teams share venues with limited capacity. Davari et al. [7] consider the problem of scheduling multiple leagues where interdependencies arise due to the clubs with teams in different leagues sharing the same venue. The leagues are assumed to be of the same size. In a consequent work, Li et al. [19] introduce a general multi-league sports scheduling problem where timetables for multiple leagues must be determined simultaneously when the leagues are allowed to have different sizes. Li and Goossens [20] solve an integrated multi-league problem where teams' grouping and the leagues' scheduling are simultaneously of concern. Durán et al. [13] schedule multiple youth leagues in Argentina, specifically paying attention to travel distances. While there is some work on multi-league timetabling, the literature on assigning referees to the games in multiple leagues is very sparse with the only work we could find being Battistutta et al. [3] where the authors develop a simulated annealing heuristic for a basketball league with up to three divisions and a few hundred games focusing on minimizing the referee travel times. The problem they consider bears similarities to the Traveling Umpire Problem. To the best of our knowledge, Battistutta et al. [3] is the only prior study addressing referee assignment across multiple leagues, and our work differs from it in several key respects. First, their objective minimizes total referee travel time, whereas MRAP minimizes the mismatch between game difficulty and referee performance while embedding workload fairness and mandatory rest considerations. Second, they employ a simulated annealing metaheuristic, whereas we propose an exact integer programming formulation complemented by decomposition-based heuristics whose subproblems are solved to optimality. Third, their study considers a basketball league with up to three divisions and a few hundred games, while our experiments scale to instances with 100–200 leagues. MRAP is thus, to our knowledge, the first model to capture the multi-league referee assignment problem at this level of generality, serving as the referee-assignment counterpart to the multi-league timetabling stream of [7, 19, 20].

3. Mathematical formulation

The MRAP can be informally described as follows. Given a set of games organized across multiple concurrent leagues over several rounds, and a pool of referees with heterogeneous qualifications, availabilities and performance scores, the goal is to assign exactly one referee to each game such that (i) referees officiate games whose difficulty matches their qualification level; (ii) workload is distributed fairly across referees both globally and within each league; (iii) referees are not over-exposed to particular teams or home venues; and (iv) mandatory rest policies are respected. When no eligible real referee is available, a dummy assignment is made – flagging cases that require manual intervention.

Next, we present a mathematical formulation of the MRAP. We consider a set of games, G, to which central referees, R, must be assigned. These games are played among a set of teams, T, and may belong to different rounds, denoted by W, and different leagues, denoted by L. In addition to real referees, we include a dummy referee with infinite officiating capacity, who is assigned to a game when no eligible real referee is available for any reason. The timetable of games is assumed to be known in advance. Games are scheduled in a set of predetermined time slots, which represent specific intervals during days (D) of each round when matches can be played. We use the term *slot* rather than *day* to emphasize the precise timing of a game in a round. A referee may officiate multiple games in a single day, provided that each game occurs in a different slot. Therefore, we let S be the set of time slots in a round. The sets, parameters, and decision variables used in the formulation are defined as follows.

3.1. Sets

G : games, indexed by $g, g \in \{1, ..., |G|\}$.

R : referees, indexed by $r, r \in \{1, ..., |R|\}$.

T : teams, indexed by $t, t \in \{1, ..., |T|\}$.

W : rounds, indexed by $w, w \in \{1, ..., |W|\}$.

L : leagues, indexed by $l, l \in \{1, ..., |L|\}$.

D : days of a round, indexed by $d, d \in \{1, ..., |D|\}$.

S : slots in the tournament, indexed by $s, s \in \{1, ..., |S|\}$.

$G_{t,t'}$: games played between opponents t and t'.

3.2. Parameters

$A_{g,r}$: 1 if referee r can be assigned to game g; 0 otherwise.

C : the number of rounds a referee has to wait before calling a game of the same team.

$D_{g,d}$: 1 if game g is scheduled on the d-th day of a round; 0 otherwise.

$D^{\max}$: the maximum number of games a referee can call in one day.

$G_r^{\min}$: the minimum number of games referee r should call during the tournament. Default = 0.

$G_r^{\max}$: the maximum number of games referee r can call during the tournament.

$H_{g,t}$: 1 if game g is a home game for team t; 0 otherwise.

$H_{r,t}^{\max}$: the maximum number of home games referee r can call for team t during the tournament.

$L_{g,l}$: 1 if game g is part of league l; 0 otherwise.

$L_{l,r}^{\min}$: the minimum number of games referee r should officiate in league l. Default = 0.

$L_{l,r}^{\max}$: the maximum number of games referee r should officiate in league l.

M_1, M_2 : sufficiently large numbers.

MW : the number of rounds during which a referee should rest at least one round.

$P_{g,r}$: the positive difference between game g's difficulty score and referee r's expected performance score; it equals zero when the referee's expected performance meets or exceeds the game's difficulty.

$S_{g,r,s}$: 1 if game g in slot s can be called by referee r; 0 otherwise.

$T_{g,t}$: 1 if team t is playing in game g; 0 otherwise.

$T_{r,t}^{\min}$: the minimum number of games referee r should call for team t during the tournament. Default = 0.

$T_{r,t}^{\max}$: the maximum number of games referee r can call for team t during the tournament.

$W_{g,w}$: 1 if game g is scheduled in round w; 0 otherwise.

3.3. Decision variables

d_g : 1 if game g is officiated by the dummy referee; 0 otherwise.

$o_{r,w}$: 1 if referee r officiates a game in round w; 0 otherwise.

$x_{g,r}$: 1 if game g is called by referee r; 0 otherwise.

3.4. Mathematical model (MRAP)

The integer programming formulation of the MRAP is as follows.

$$\min \sum_{g\in G}\sum_{r\in R} P_{g,r}\cdot x_{g,r} + \sum_{g\in G} M_1 \cdot d_g \tag{1}$$

$$\text{s.t.} \sum_{r\in R} A_{g,r}\cdot x_{g,r} + d_g = 1 \quad g\in G \tag{2}$$

$$\sum_{g\in G} x_{g,r} \geq G_r^{\min} \quad r\in R \tag{3}$$

$$\sum_{g\in G} x_{g,r} \leq G_r^{\max} \quad r\in R \tag{4}$$

$$\sum_{g\in G} L_{g,l}\cdot x_{g,r} \geq L_{l,r}^{\min} \quad r\in R, l\in L \tag{5}$$

$$\sum_{g\in G} L_{g,l}\cdot x_{g,r} \leq L_{l,r}^{\max} \quad r\in R, l\in L \tag{6}$$

$$\sum_{g\in G} S_{g,r,s}\cdot x_{g,r} \leq 1 \quad r\in R, s\in S \tag{7}$$

$$\sum_{g\in G} D_{g,d}\cdot x_{g,r} \leq D^{\max} \quad r\in R, d\in D \tag{8}$$

$$\sum_{g\in G} T_{g,t}\cdot x_{g,r} \geq T_{r,t}^{\min} \quad r\in R, t\in T \tag{9}$$

$$\sum_{g\in G} T_{g,t}\cdot x_{g,r} \leq T_{r,t}^{\max} \quad r\in R, t\in T \tag{10}$$

$$\sum_{g\in G} H_{g,t}\cdot x_{g,r} \leq H_{r,t}^{\max} \quad r\in R, t\in T \tag{11}$$

$$\sum_{g\in G} T_{g,t}\cdot \sum_{j=0}^{C} W_{g,(w+j)}\cdot x_{g,r} \leq 1 \quad r\in R, t\in T, w\in W \tag{12}$$

$$x_{g,r} + x_{g',r} \leq 1 \quad r\in R, g\in G_{t,t'}, g'\in G_{t,t'}, g\neq g', t\in T, t'\in T, t\neq t' \tag{13}$$

$$\sum_{g\in G} W_{g,w}\cdot x_{g,r} \leq M_2\cdot o_{r,w} \quad r\in R, w\in W \tag{14}$$

$$\sum_{w=w'}^{w'+MW-1} o_{r,w} \leq MW-1 \quad r\in R, w'\in\{1,...,|W|-MW+1\} \tag{15}$$

$$d_g, o_{r,w}, x_{g,r}\in\{0,1\} \quad g\in G, r\in R, w\in W \tag{16}$$

Objective function

Equation (1) minimizes an objective function consisting of two components. The first component penalizes mismatches between the difficulty level of each game and the seasonal expected performance score of the assigned referee. This ensures that games are officiated by referees whose competence is appropriately aligned with the game's demands. The mismatch penalty, denoted by $P_{g,r}$, is the positive difference between the difficulty score of game g and the performance score of referee r. Thus, no penalty is incurred when a referee's expected performance exceeds the difficulty of the assigned game.

Both game difficulty and referee performance can be reasonably estimated in a non-professional context. Game difficulty may reflect league tier, team rivalry, or timing within the season (e.g., more difficult if played towards the end of the season), while referee performance scores (that will apply to the upcoming season) can be based on past evaluations, participation in training seminars, and physical assessments. Minimizing the total mismatch penalty therefore promotes skill-aligned assignments. Furthermore, the $P_{g,r}$ values can serve as a practical guide for prioritizing or avoiding some assignments, for example, discouraging the appointment of inexperienced referees to high-stakes matches.

The second component penalizes assignments to the dummy referee, who serves as a placeholder when no eligible real referee can be assigned. Each such assignment incurs a penalty weighted by a large constant M_1, ensuring that avoiding dummy assignments is prioritized over minimizing mismatch penalties. The presence of many dummy assignments, especially when an adequate number of referees is available, may indicate that the assignment constraints are overly restrictive and require relaxation or manual override.

Assignment constraints

Constraints (2) are soft constraints so that each game is assigned a referee. If this is not possible for a game, the dummy referee is assigned to that game. Games assigned to the dummy referee will need manual intervention after the optimization. On the other hand, the parameter $A_{g,r}$ can be set accordingly if certain referees are not available/eligible to officiate some games for any reason. For example, they may not be available on the date of a game due to personal reasons, or they may not be eligible to call games in a certain league.

Seasonal limitations

Constraints (3) and (4) set lower and upper bounds on the number of games referee r can officiate throughout the tournament. Such constraints may be needed to balance the overall workload of the referees. Note that the referees are paid based on their workload. Therefore, these constraints also help achieve fairness in payments, aside from easing the physical burden on the referees. For example, a federation may impose a referee-specific limit to officiate no more than $G_r^{\max}$ and no less than $G_r^{\min}$ games per season to prevent overloading and burnout of referees while ensuring that every referee gets fair opportunities. Moreover, more experienced referees may have different limits compared to new referees.

Limitations regarding leagues

Constraints (5) and (6), on the other hand, set lower and upper limits for the referees on the number of games to officiate in each league. Observe that when a referee is not eligible to call games in a particular league, this can be accomplished in two ways in the model: by setting the related $A_{g,r}$ or the respective $L_{l,r}^{\min}$ and $L_{l,r}^{\max}$ values to 0. On the other hand, when a referee is eligible to call games in a league, one may still need to control the number of games a referee should call in that league which can be accomplished with Constraints (5) and (6). For example, these constraints can be used when a junior referee should gain experience across different leagues but should not be overwhelmed with too many higher-level games.

Daily limitations

The next couple of constraints are for daily limitations for each referee. Constraints (7) enforce each referee to call at most one game in a given slot since a referee cannot physically officiate two games at the same time, even if they are within the same venue. By utilizing $S_{g,r,s}$ coefficients, one can restrict the slots a referee has games in. Such needs may arise, for example, when a referee is not available during certain times in a day coinciding with a certain slot, or when, due to travel distances, a referee cannot get to the games in a slot if s/he has already been pre-assigned to a game in another slot. Constraints (8) limit the number of games a referee can officiate per day to $D^{\max}$ to ensure performance quality. For example, officiating several games on the same day could lead to fatigue, affecting decision-making.

Limitations regarding teams

There are several constraints related to referees being eligible to call games of specific teams. Constraints (9) and (10) let each referee r call at least $T_{r,t}^{\min}$ and at most $T_{r,t}^{\max}$ games for team t, respectively. Since a referee who officiates multiple games for the same team may develop unconscious biases, these constraints ensure that referees do not officiate too many games for a single team, avoiding favoritism or bias. Constraints (11) set an upper bound for the number of home games of a team to be officiated by the same referee to prevent familiarity biases. This number is limited because a referee's decisions may be influenced by being in front of the same crowd too frequently. Constraints (12) prevent referees from calling consecutive games for the same team, reducing the risk of bias. After how many rounds a referee can call a game of a team depends on the C value. Many leagues play a double round-robin tournament where each pair of teams will face each other twice. Thus, Constraints (13) enforce that referees do not officiate games between the same opponents more than once. This prevents any referee from becoming overly involved in the outcomes of a particular rivalry.

Limitations regarding rests

Constraints (14) are switching constraints for capturing in which weeks (rounds) a referee is officiating. Constraints (15) oblige referees to rest for at least one round in every MW consecutive rounds. Namely, these constraints require referees to take mandatory breaks after officiating a certain number of rounds to prevent fatigue and ensure they remain sharp.
Constraints (16) identify the type of the decision variables as binary.

Notes

If the importance of games is to be differentiated by leagues, this can be reflected by setting a high value for the more important games. As a result, the related $P_{g,r}$ will change. While $A_{g,r}$ values dictate referee eligibility, $P_{g,r}$ values are soft and allow assignments with a penalty providing a prioritization mechanism. It should be also noted that, for notational simplicity, the formulation assumes uniform parameter values across leagues for certain constraints, such as the maximum number of games a referee may officiate for the same team ($T_{r,t}^{\max}$). However, the model is readily extensible to accommodate league-specific parameterization. By indexing these parameters with an additional league dimension, practitioners can impose differentiated policies that reflect the distinct operational requirements or competitive characteristics of each league.

4. Heuristic methods

When realistically-sized problem instances cannot be solved to optimality within reasonable computational time, or hit memory limitations, decomposition strategies offer a practical alternative. For example, mid-season re-optimization due to factors such as postponements and changes in referee availability is expected to be common in practice, and it would require rapid turnaround. The heuristic methods described in this section partition the full problem using two approaches. The first approach employs temporal decomposition: the optimization is performed for all leagues over a fixed planning horizon of consecutive rounds, where the optimization window moves forward by the fixed horizon. The second approach applies structural decomposition: leagues are partitioned into batches and optimized sequentially for the entire season, typically following a top-down hierarchy based on competitive tier or age group in non-professional sports contexts.

Both decomposition strategies require careful management of global constraints across sequential subproblems. To ensure consistency, we track cumulative assignments from previously solved subproblems and adjust constraints accordingly. For example, if a referee has officiated games in three consecutive rounds immediately preceding the current subproblem, and the rest policy requires at least one break every four rounds, that referee is prohibited from being assigned in the first round of the current subproblem. Similar enforcement mechanisms are applied to prevent consecutive assignments involving the same teams. Furthermore, resource constraints (such as the maximum number of games a referee may officiate throughout the season) are dynamically adjusted by subtracting previously committed assignments, ensuring that cumulative limits are respected across all subproblems.

As demonstrated in our numerical experiments (Section 5), the monolithic formulation becomes computationally intractable for realistically-sized instances due to excessive runtime or memory requirements. Although the RAP is NP-hard, the decomposed subproblems are sufficiently small to be solved exactly and efficiently using commercial solvers. This makes the proposed heuristics highly suitable for practical implementation, as they combine the rigor of exact optimization with the scalability advantages of problem decomposition. While further refinement of the partitioning scheme is theoretically possible (for instance, by optimizing fewer leagues in each subproblem of the first approach, or by using shorter time horizons in the second approach) our computational results suggest that such additional decomposition is unnecessary, as the current approaches already achieve acceptable solution times.

4.1. Batching by weeks

This section describes the temporal decomposition heuristic that partitions the problem by rounds. The pseudocode is provided in Algorithm 1. The algorithm operates by iteratively solving a sequence of subproblems, where each subproblem encompasses all leagues over a fixed batch of u consecutive rounds. After solving each subproblem, the window advances by u rounds. Beginning with rounds 1 through u, the algorithm then proceeds to rounds $u+1$ through $2u$, and so forth. When fewer than u rounds remain in the season, the final subproblem optimizes all remaining rounds. Between consecutive subproblems, the model parameters are updated to incorporate the outcomes of prior assignments (including cumulative referee workloads, rest period history, and team-specific assignment counts) thereby maintaining global constraint consistency throughout the sequential optimization process. Note that this is not a rolling horizon approach: there are no overlapping optimizations, and each round belongs to exactly one batch. The optimization window simply advances by u rounds without revisiting previously solved batches.

Set $|W|$ = Number of rounds (weeks) in the full season
Set $|L|$ = Number of leagues
Set u = Desired batch length (number of weeks)
for $i \in \{0, 1, ..., \lceil |W|/u \rceil - 1\}$ **do**
    $w = 1 + i \cdot u$
    $optHrzn = \min\{u, |W| - w + 1\}$
    Solve MRAP for $optHrzn$ weeks from w to $w + optHrzn - 1$ for $|L|$ leagues
    Update model parameters based on assignments from week w to week $w + optHrzn - 1$
end for

Algorithm 1. Pseudocode for weekly batching

A key design question in cross-batch constraint management concerns how seasonal upper bounds (e.g., $G_r^{\max}$, $T_{r,t}^{\max}$, $H_{r,t}^{\max}$, $L_{l,r}^{\max}$) are handled across batches. In our implementation, the first batch starts with the global upper bounds as given, and each subsequent batch has its effective upper bound reduced by the number of assignments already committed in preceding batches. While this gives the first batch full flexibility and may risk depleting capacity for later batches, our tight $G_r^{\max}$ setting (set to $\lceil$total games$/|R|\rceil$ to enforce equitable workload) naturally acts as a pro-rata capacity guard: referees quickly exhaust their quota once it is reached, preventing any single early batch from monopolizing their time. This is a key reason why our experimental results do not suffer noticeably from the myopic first-batch advantage.

The situation is different for lower bounds ($G_r^{\min}$, $L_{l,r}^{\min}$, $T_{r,t}^{\min}$). Enforcing large lower bounds in early batches is clearly counterproductive, as it forces all demand to be met quickly and severely constrains subsequent batches. A more appropriate strategy is to pro-rate lower bounds across batches in proportion to the batch's share of the season: for instance, if a batch covers 3 out of 11 rounds, a referee's minimum games for that batch is set to $\lfloor G_r^{\min} \cdot 3/11 \rfloor$. This ensures minimum requirements are spread evenly, avoiding infeasibility in later batches. In our numerical experiments, all lower bounds are set to zero, so this issue does not arise in practice. However, any practical deployment with non-zero lower bounds should adopt a pro-rata apportionment strategy. The counterintuitive result that shorter temporal horizons ($u = 1$) outperform longer ones ($u = 3$ or $u = 5$) encountered during numerical experiments is also partially explained by this cross-batch interaction: longer horizons commit more capacity in each subproblem, leaving less flexibility for subsequent subproblems and causing solution quality to deteriorate. The league batching heuristic described next is less sensitive to this phenomenon because the leagues within a batch are largely independent of one another (they share only the referee pool), whereas temporal batches always share both referee pools and constraint histories.

4.2. Batching by leagues

The second heuristic method employs structural decomposition by sequentially assigning referees to leagues in hierarchical order, beginning with the highest-priority league and proceeding downward. The pseudocode is provided in Algorithm 2. Rather than optimizing all leagues simultaneously, this approach processes leagues in batches of size m, ordered by importance (typically determined by competitive tier, age group, or league level). The algorithm iteratively optimizes referee assignments for m leagues at a time across the entire season. Following the optimization of each batch, model parameters are updated to account for the committed assignments, thereby reducing the available capacity for subsequent batches.

Set $|W|$ = Number of rounds (weeks) in the full season
Set $|L|$ = Number of leagues
Set m = Number of leagues to be optimized together
for $i \in \{0, 1, \ldots, \lceil |L|/m \rceil - 1\}$ **do**
  $l = 1 + i \cdot m$
  $optLg = \min\{m, |L| - l + 1\}$
  Solve MRAP for $optLg$ leagues from l to $l + optLg - 1$ for $|W|$ weeks
  Update model parameters based on the assignment decisions of leagues l to $l + optLg - 1$
end for

Algorithm 2. Pseudocode for batching by leagues

The cross-batch constraint management discussed in Section 4.1 for temporal decomposition does not require the same level of intricacy in the league batching heuristic. Since each subproblem optimizes the entire tournament for its assigned leagues, there is no temporal fragmentation of constraints: all intra-tournament requirements ($T^{\max}r, t$, $H^{\max}r, t$, rest periods, etc.) are enforced in full within each subproblem. The only cross-batch interaction arises from the shared referee pool: after each league batch commits its assignments, the global seasonal bounds ($G^{\max}r$, $L^{\max}l, r$) are reduced by the number of games already assigned. Because there is no need to apportion these bounds over time, the pro-rata lower-bound issue that arises in weekly batching (Section 4.1) does not apply here. The remaining challenge – that early league batches may deplete high-quality referees for lower-priority leagues – is an inherent property of the hierarchical ordering rather than a constraint management question.

5. Numerical experiments

5.1. Problem instances and setup

Numerical experiments were executed on an Intel Core i7-1165G7 CPU 2.8 GHz computer with 16GB of RAM using Gurobi 9.5.1 as the solver and Python as the modeling environment. In our mathematical model, the mismatch penalty for a game is determined by comparing a referee's expected seasonal performance to the difficulty level of the game they are assigned to. Referee performance scores were assumed to lie in the range of 7.4 to 9.5, in increments of 0.1. This range reflects typical evaluation systems employed by many football federations [26], where referees are rated on a 10-point scale but rarely receive scores below 7.0, as consistently underperforming referees are generally excluded from the assignment pool. The upper limit of 9.5 was selected to represent only the most exceptional referees, thereby producing a realistic distribution of performance scores. Referee performance scores were assumed to be estimated at the beginning of the season and held constant across all weeks. Game difficulty levels were discretized into three categories: normal, hard, and very hard, with corresponding scores of 7.4, 8.1, and 8.8, respectively. Given that referee scores were generated uniformly at random over the 7.4–9.5 interval, a normal game is callable by all referees, a hard game is callable by approximately two-thirds of referees, and a very hard game is callable by only about one-third.

All league schedules were assumed to follow a single round-robin format, and not a double round-robin format, primarily to obtain the numerical results more quickly during our experiments. Moreover, in practice, it may be more desirable to run an additional optimization before the second half of a double round-robin tournament, so that assignments better reflect changes in referee availabilities and performance scores that emerge during the first half. The fixture for each league, including home and away designations, was generated canonically using the procedure described in [8]. These fixtures directly specify the binary parameter values for $T_{g,t}$ (team participation in games), $H_{g,t}$ (home team indicators), and $W_{g,w}$ (scheduled round of games). $L_{g,l}$ values, league affiliations of games, are part of the input data determined by considering which teams compete in which league and the games they will play among each other. Since each pair of teams plays exactly once in a single round-robin setting (i.e., $|G_{t,t'}| = 1$ for all team pairs t, t'), Constraints (13) became redundant and were omitted from the model (except a few experiments reported in Table 2).

We also assumed that each round (week) was composed of $|D| = 2$ matchdays each with one slot (thus $|S| = |W| \cdot |D|$), and games were distributed across these days to minimize the total rest difference between opposing teams. Specifically, teams were scheduled so that opponents had equal or nearly equal rest durations following their games in the previous round, as described in [5]. These matchday assignments directly define the values of the binary parameters $D_{g,d}$. Since we assumed that there is one slot on every day, each referee can call at most one game (Constraints (7)) on a given day ($D^{\max} = 1$).

In the experiments, the number of referees was set as a multiple (λ) of the total number of weekly games, i.e., $|R| = \lambda \cdot (|T|/2)$. The maximum number of games a referee can officiate ($G_r^{\max}$) was calculated by dividing the total number of games by the number of referees and rounding up to the nearest integer. This setting ensures a uniform workload distribution among referees. Unless a particular experiment aimed to test the sensitivity of a specific parameter, the following values were fixed throughout: the referee coefficient λ was set to 1.2; referees were required to rest at least once every $MW = 4$ rounds; and each referee could officiate at most $T_{r,t}^{\max} = 4$ games for any team, including at most $H_{r,t}^{\max} = 3$ home games. Except where stated otherwise, the number of games a referee could call within the same league ($L_{l,r}^{\max}$) was left unconstrained, meaning that all of a referee's assignments could be from a single league. All minimum thresholds ($G_r^{\min}$, $L_{l,r}^{\min}$, and $T_{r,t}^{\min}$) were set to zero. Additionally, referees were prohibited from officiating consecutive games of the same team by setting $C = 1$. All referees were assumed to be eligible for all games by assigning $A_{g,r} = 1$ for all (g, r) pairs. We did not randomly restrict eligibility by setting some $A_{g,r}$ values to zero, as referee suitability was already encoded through the randomly generated mismatch penalties in the experiments. In practice, however, the $A_{g,r}$ parameter allows system administrators to restrict assignments based on referee availability, eligibility, or administrative rules. Along a similar reasoning, we also assumed that all referees are available in all slots ($S_{g,r,s} = 1$).

The results reported in the tables below represent *averages* over 10 independent instances. Each instance has a different set of randomly generated referee performance scores, with the default game difficulty distribution set to 80% normal, 15% hard, and 5% very hard games, unless otherwise stated.

5.2. Monolithic model

Table 1 gives the number of variables and constraints with different numbers of leagues with 12 teams each and $\lambda = 1.2$ showing the quick increase both in the number of variables and constraints for the league sizes we report results for.

Table 1. Number of variables and constraints.

Number of leagues	10	50	100	200
Number of variables	48,972	1,119,260	4,766,520	19,037,040
Number of constraints	117,516	2,862,780	11,413,560	45,579,120

Table 2 tabulates the runtimes of solving MRAP as a monolithic model for the whole season where the number of leagues ($|L|$) is 10 and 50, and each league had 12 teams. In this and all subsequent tables, the column labeled *Dummy assignments* reports the average number of games left unassigned to a real referee. *Games under par* indicates the average number of games where the assigned referee's performance score falls below the game's difficulty score. The total of such performance gaps, on average, is provided under *Total under par*. *Referees used* denotes the average number of referees assigned across instances. Since the maximum number of games per referee is typically set to ensure an equitable distribution of workload, in most cases, all referees are expected to be utilized. Finally, *Min. games* and *Max. games* show the average minimum and maximum number of games officiated by any single referee, respectively, capturing the spread of workload across the referee pool.

Table 2. MRAP runtimes.

Leagues	Teams	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
10	120	2.24	0.00	0.00	0.00	72.00	2.80	10.00
10 (2RR)	120	9.49	0.00	0.00	0.00	72.00	9.40	20.00
50	600	179.77	0.00	0.00	0.00	359.90	3.20	10.00
50 (2RR)	600	698.08	0.00	0.00	0.00	360.00	9.60	20.00
50 (100 secs.)	600	101.30	3300.00	3300.00	25,000.79	0.00	0.00	0.00
50 (150 secs.)	600	150.90	990.00	990.00	7,500.09	252.00	2.20	7.00

The results in Table 2 indicate that the monolithic model efficiently solves the problem when the instance size is small (10 leagues, 120 teams and 72 referees), requiring only a few seconds. However, as the number of leagues increases to 50 (with 600 total teams and 360 referees), the running time surges approximately eightyfold, demonstrating the substantial growth in computational complexity. We also solved the monolithic problem as a double round-robin tournament (2RR) for these problem sizes. In 2RR experiments, the maximum number of games for a team by each referee was doubled. The minimum number of games that each referee calls approximately triples, while the runtimes increase roughly fourfold, showing the impact of Constraints (13). When further challenged with 100 leagues, even a short 200-second time limit yielded only trivial all-dummy solutions with an infinite optimality gap (the lower bound remained at zero), and longer runs exhausted all available memory. This highlights that solving the monolithic model for a large number of non-professional leagues and teams is impractical without significantly more computational resources. This underscores the monolithic approach's lack of scalability for large instances, making it infeasible for real-world applications involving hundreds of leagues. Despite these computational limitations, the model consistently produced high-quality assignments, as evidenced by a Dummy assignments count of zero across all instances solved to optimality, ensuring that every game received a referee. Moreover, Games under par and Total under par remained at zero in these instances, indicating that all referees met or exceeded the required performance score for their assigned games. The referee utilization was also near-ideal, with all referees being assigned in every instance except one 50-league scenario, where the average utilization across ten instances was 359.90 referees. Additionally, the minimum and maximum number of games assigned per referee were well-balanced, reinforcing that workload distribution and fairness constraints were effectively enforced by the model. We also checked the quality of solutions under time limits as reported in Table 2 for 50 leagues. The average optimality gap was 99.95% with a time limit of 100 seconds, which reduced to 30.04% when the time limit was increased to 150 seconds (close to when optimality was found on average). Considering that optimality was proved soon after, early termination of the monolithic approach was not promising as a heuristic. We further note that the optimal objective value is zero in the baseline instances: since all minimum thresholds are set to zero and every referee is eligible for every game ($A_{g,r} = 1$), each game can be matched to a referee whose expected performance meets or exceeds its difficulty, so no penalty is incurred. Consequently, the linear programming relaxation also attains a zero optimum and the duality gap is zero. Strictly positive objective values arise only in more demanding settings–such as the heavily skewed game-difficulty distributions and the reduced-availability experiments reported below–where exact difficulty–performance matching is no longer attainable.

5.3. Performance of heuristic methods

Table 3 unifies the comparison of the monolithic model and both heuristic approaches across the instance sizes for which the monolithic model is still tractable (10 leagues with $\lambda = 1.0$, and 50 leagues with $\lambda = 1.2$). The upper panel covers the 10-league instances and evaluates the weekly batching heuristic with batch sizes $u = 1, 3, 5$ and the league batching heuristic with batch sizes $m = 1, 2, 5$, all against the monolithic baseline. The lower panel covers the 50-league instances and evaluates weekly batching with $u = 1, 5$ and league batching with $m = 1, 25$.

Across both instance sizes, all heuristic variants achieve dramatically lower runtimes than the monolithic model. For 10-league instances, the weekly batching heuristic with $u = 1$ produces the best solution quality among heuristics, with 35 games under par; increasing the batch size to $u = 3$ or $u = 5$ worsens quality despite longer runtimes, confirming the counterintuitive result that shorter batches outperform longer ones. League batching yields no dummy assignments and stable quality regardless of batch size, because the full season is optimized within each subproblem. For 50-league instances, weekly batching with $u = 1$ again dominates in quality, while $u = 5$ introduces substantial dummy assignments and penalties. League batching for 50 leagues consistently produces far more under-par assignments than weekly batching with $u = 1$, because high-quality referees assigned to early league batches are depleted for later batches. Batch size has minimal effect on league batching quality in both panels, confirming that the hierarchical ordering of leagues – not decomposition granularity – drives the performance gap. Further stress tests of both heuristics under reduced referee availability and non-zero minimum workload requirements are reported in Table 5.

Table 4 further evaluates the heuristics on instances large enough that the monolithic model exhausts memory (100 and 200 leagues). Both heuristics remain tractable at these scales. Weekly batching with $u = 1$ consistently produces the best quality with no dummy assignments, while increasing to $u = 2$ introduces heavy dummy assignments and quality penalties. League batching exhibits more stable quality across batch sizes but produces substantially higher under-par counts than weekly batching with $u = 1$, confirming the pattern observed in Table 3. The stability of league batching results confirms that cross-batch constraint management is straightforward in the league batching setting as discussed in Section 4.2.

Table 3. Monolithic model vs. heuristic approaches: 10-league ($\lambda = 1.0$) and 50-league ($\lambda = 1.2$) instances.

Instance	Method	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
10 leagues, 12 teams/league, 60 referees, $\lambda = 1.0$								
	Monolithic	585.74	0.00	0.00	0.00	60.00	11.00	11.00
	Weekly batching, $u = 1$	0.47	30.00	35.00	230.52	60.00	10.00	11.00
	Weekly batching, $u = 3$	0.76	43.60	56.50	339.18	60.00	7.80	11.00
	Weekly batching, $u = 5$	78.46	40.43	62.43	315.19	60.00	8.29	11.00
	League batching, $m = 1$	2.16	0.00	51.30	26.20	60.00	11.00	11.00
	League batching, $m = 2$	2.96	0.00	51.20	26.23	60.00	11.00	11.00
	League batching, $m = 5$	9.43	0.00	50.70	26.18	60.00	11.00	11.00
50 leagues, 12 teams/league, 360 referees, $\lambda = 1.2$								
	Monolithic	179.77	0.00	0.00	0.00	359.90	3.20	10.00
	Weekly batching, $u = 1$	17.36	0.00	22.40	11.18	360.00	4.20	10.00
	Weekly batching, $u = 5$	97.05	89.00	203.20	736.86	360.00	4.00	10.00
	League batching, $m = 1$	53.35	0.00	260.10	141.74	357.90	0.20	10.00
	League batching, $m = 25$	171.29	0.00	259.00	139.19	360.00	2.80	10.00

Table 5 provides stress tests of both heuristics for 10-league instances ($\lambda = 1.0$), examining the effects of cross-batch capacity management, non-zero workload bounds, and reduced referee availability. The upper panel covers the weekly batching heuristic and the lower panel covers the league batching heuristic. The first block of each panel repeats the baseline results from Table 3 for reference.

The second block of the upper panel investigates the effect of per-batch upper bounds ($G_r^{\max}$) on solution quality for weekly batching. In the default setting, $G_r^{\max}$ applies to the whole season; in these experiments we instead impose a per-batch cap, specified in parentheses as a sequence of values across

Table 4. Performance of heuristic approaches for realistic sizes (monolithic infeasible due to memory)

Method (Leagues, Teams, Referees)	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
Weekly batching, $u = 1$ (100,1200,720)	84.91	0.00	47.10	26.21	720.00	3.60	10.00
Weekly batching, $u = 2$ (100,1200,720)	199.97	195.33	287.00	1533.20	720.00	4.00	10.00
League batching, $m = 10$ (100,1200,720)	247.74	0.00	512.40	274.40	718.30	1.40	10.00
League batching, $m = 20$ (100,1200,720)	466.17	0.00	522.67	277.90	716.67	0.67	10.00
Weekly batching, $u = 1$ (200,2400,1440)	449.91	0.00	94.33	51.77	1440.00	3.33	10.00
Weekly batching, $u = 2$ (200,2400,1440)	1612.76	590.00	767.00	4565.03	1440.00	4.00	10.00
League batching, $m = 10$ (200,2400,1440)	1217.27	0.00	1009.00	541.00	1406.00	0.00	10.00
League batching, $m = 20$ (200,2400,1440)	1935.52	0.00	1032.67	549.70	1429.67	2.00	10.00

the batches. The extreme case of restricting every referee to at most one game per batch ("all $G_r^{\max}$ 1" with $u = 1$) creates a completely balanced load in each round but results in a dramatic deterioration of solution quality, with 300 dummy assignments and more than 305 games under par. This is because such a tight constraint leaves very little room to cover all games with qualified referees in every single round. When the per-batch caps are distributed more evenly (e.g., $G_r^{\max}$ = 3,3,3,2 across four batches for $u = 3$), the degradation is more moderate. Front-loading the per-batch capacity (e.g., $G_r^{\max}$ = 2,2,3,4 for $u = 3$, or $G_r^{\max}$ = 4,6,1 for $u = 5$) considerably worsens quality compared to back-loading, confirming that reserving capacity for later batches is beneficial. These results validate the cross-batch management discussion in Section 4: the default season-wide $G_r^{\max}$ with residual subtraction naturally produces a near-pro-rata distribution and yields the best outcomes among these variants.

The third block of the upper panel explores the impact of non-zero lower bounds ($G_r^{\min}$) on weekly batching. When the minimum is set to 1 for referees assigned in shorter batches and 2 otherwise ("$G_r^{\min}$ 1 $G_r^{\max}$ 11,13"), feasibility is restored but solution quality worsens compared to the zero-lower-bound case. With longer horizons ($u = 3$ or $u = 5$) and $G_r^{\min} = 2$ applied per batch (approximately proportional to tournament share), the lower bound becomes achievable and solution quality actually improves over the zero-lower-bound variants with the same horizon, since the constraint helps the solver avoid trivially assigning too few games to referees in early batches. This confirms that a pro-rata apportionment of lower bounds across batches, as outlined in Section 4, is the appropriate strategy when $G_r^{\min} > 0$.

The fourth block of the upper panel reports weekly batching results when referee availability is aggressively reduced to 50% (i.e., each referee is eligible for each game independently with probability 0.5). For $u = 1$ and $u = 3$, the runtimes are comparable to or shorter than the fully available case, because the reduced eligibility shrinks the feasible solution space and simplifies some assignment decisions. The solution quality, however, is similar to the fully available case: the same horizon-length ordering ($u = 1$ best, $u = 3$ worst) holds. This demonstrates that the heuristic is robust to moderate levels of referee unavailability. The $u = 5$ case required a 2% MIP gap tolerance to terminate within a reasonable time, yet still produced results comparable to the default $u = 5$ outcome.

The lower panel presents the analogous stress tests for league batching. The second block shows that introducing a minimum game requirement ($G_r^{\min} = 2$, $G_r^{\max} = 13$) has virtually no adverse effect on solution quality: dummy assignments remain at zero and games under par are essentially unchanged relative to the baseline. This is because league batching optimizes the entire season within each subproblem, so the global $G_r^{\min}$ bound is naturally enforceable in a single solve without the cross-batch fragmentation that causes dummy assignments in weekly batching under the same setting. The third block confirms that league batching is similarly robust under 50% referee availability: dummy assignments remain at zero across all batch sizes, solution quality is nearly unchanged, and – in contrast to the fully available case – runtimes increase substantially with batch size, reflecting the added combinatorial difficulty of covering all games when the eligible referee pool is halved.

Table 5. Stress tests for both heuristics: cross-batch management, workload bounds, and referee availability (10 leagues, 12 teams/league, 60 referees, $\lambda = 1.0$).

Method / Setting	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
Monolithic (baseline)	585.74	0.00	0.00	0.00	60.00	11.00	11.00
Weekly batching							
$u = 1$	0.47	30.00	35.00	230.52	60.00	10.00	11.00
$u = 3$	0.76	43.60	56.50	339.18	60.00	7.80	11.00
$u = 5$	78.46	40.43	62.43	315.19	60.00	8.29	11.00
$u = 1$ (all $G_r^{\max} = 1$)	0.40	300.00	305.80	2,273.97	60.00	6.00	6.00
$u = 3$ ($G_r^{\max} = 3, 3, 3, 2$)	4.29	62.00	76.60	480.10	60.00	7.50	11.00
$u = 3$ ($G_r^{\max} = 2, 2, 3, 4$)	2.09	158.80	163.50	1,212.15	60.00	7.00	11.00
$u = 5$ ($G_r^{\max} = 5, 5, 1$)	59.80	56.80	81.00	444.04	60.00	9.00	11.00
$u = 5$ ($G_r^{\max} = 4, 6, 1$)	1.27	98.40	109.20	765.58	60.00	5.50	11.00
$u = 1$ ($G_r^{\min} = 1$, $G_r^{\max} = 11, 13$)	0.35	60.00	65.20	456.77	60.00	9.00	11.00
$u = 3$ ($G_r^{\min} = 2$, $G_r^{\max} = 13$)	1.15	30.50	42.60	237.94	60.00	6.00	12.70
$u = 5$ ($G_r^{\min} = 2$, $G_r^{\max} = 13$)	1.07	22.00	44.10	178.69	60.00	6.00	13.00
$u = 1$ ($P(A_{g,r} = 1) = 0.5$)	0.42	30.00	34.30	230.05	60.00	10.00	11.00
$u = 3$ ($P(A_{g,r} = 1) = 0.5$)	0.43	47.00	59.00	363.44	60.00	7.40	11.00
$u = 5$ ($P(A_{g,r} = 1) = 0.5$, MIPGap=2%)	39.70	41.70	64.70	327.87	60.00	8.00	11.00
League batching							
$m = 1$	2.16	0.00	51.30	26.20	60.00	11.00	11.00
$m = 2$	2.96	0.00	51.20	26.23	60.00	11.00	11.00
$m = 5$	9.43	0.00	50.70	26.18	60.00	11.00	11.00
$m = 2$ ($G_r^{\min} = 2$, $G_r^{\max} = 13$)	1.54	0.00	53.60	26.89	60.00	5.00	12.00
$m = 5$ ($G_r^{\min} = 2$, $G_r^{\max} = 13$)	2.16	0.00	50.70	28.45	60.00	5.40	12.00
$m = 1$ ($P(A_{g,r} = 1) = 0.5$)	1.87	0.00	52.30	29.91	60.00	11.00	11.00
$m = 2$ ($P(A_{g,r} = 1) = 0.5$)	10.01	0.00	51.60	27.40	60.00	11.00	11.00
$m = 5$ ($P(A_{g,r} = 1) = 0.5$)	123.52	0.00	52.80	27.95	60.00	11.00	11.00

5.4. Sensitivity analysis with the monolithic model

To test the sensitivity of the model to the values of the different parameters, we set the number of leagues to 10 each with 12 teams in Table 6 through Table 8. Table 6 explores how varying the number of referees, controlled via the referee coefficient λ, affects the runtime and assignment quality under the monolithic MRAP formulation. Three scenarios are tested with coefficients of $\lambda = 1.0, 1.2$, and 1.5, corresponding to increasing referee pool size. The results reveal the relationship between number of referees and runtime. When the number of referees is low ($\lambda = 1.0$), the optimization process becomes substantially more difficult, likely due to tight feasibility and limited solution space. As more referees become available, the search space expands, making it easier for the solver to find feasible and high-quality assignments. However, beyond a certain point (e.g., $\lambda = 1.5$), the benefit in runtime diminishes, due to an increase in decision variables. In all three scenarios, the Dummy assignments, Games under par, and Total under par are all zero, indicating that every game is assigned a qualified referee, and no mismatches occur between referee performance and game difficulty. This means the model consistently produces high-quality solutions regardless of referee count. This also confirms that the assignment quality is robust to the number of referees, as long as the minimum required referee pool is met. Moreover, as the referee pool increases, Referees used increases, while Min. games and Max. games decrease. This is because, with more referees available, the workload is distributed more evenly, and individual referees are assigned fewer games, supporting the fairness goal of the model. When the referee availability is restricted by 50%, the runtimes increase when the referee supply is tight ($\lambda = 1.0$). However, they decrease with increased referee supply possibly because the number of feasible solutions decrease, and there are still referees with proper qualifications available. The minimum number of games refereed increase as well.

Table 6. MRAP runtimes with 10 leagues and 12 teams each for different numbers of referees.

Referee coefficient	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
1.0	585.74	0.00	0.00	0.00	60.00	11.00	11.00
1.2	2.24	0.00	0.00	0.00	72.00	2.80	10.00
1.5	2.92	0.00	0.00	0.00	89.90	2.20	8.00
1.0 ($P(A_{g,r} = 1) = 0.5$)	761.59	0.00	0.00	0.00	60.00	11.00	11.00
1.2 ($P(A_{g,r} = 1) = 0.5$)	1.11	0.00	0.00	0.00	72.00	4.20	10.00
1.5 ($P(A_{g,r} = 1) = 0.5$)	1.80	0.00	0.00	0.00	90.00	2.40	8.00

Table 7 explores the sensitivity of the MRAP's performance to changes in the distribution of game difficulty levels while keeping the problem size and number of referees constant (10 leagues, 12 teams per league, 72 referees, $\lambda = 1.2$). The three scenarios tested represent different proportions of normal, hard, and very hard games. As expected, runtime increases slightly as the proportion of harder games rises, primarily because a greater share of games becomes callable only by a small subset of high-performing referees. This narrows the feasible solution space, thereby increasing solver effort. The MRAP yields no under-par assignments when most games are normal or moderately difficult. However, in the most demanding scenario (when 80% of games require high-performing referees), the model is forced to assign referees whose expected performance falls short of the game difficulty, resulting in under-par assignments. In this case, the number of under-par games rises substantially, and some referees are assigned very few games, with the minimum dropping to near zero. Across all difficulty distributions, the number of referees used remains equal to or very close to 72, and the spread of assignments (measured by minimum and maximum games per referee) remains relatively stable, except in the last scenario where fairness slightly deteriorates due to the difficulty of matching game requirements with available qualifications. Still, some referees are underutilized and many games are called by less qualified referees since most of the games are very hard. There are, however, no dummy assignments because such assignments are highly penalized. Overall, these results indicate that the model maintains both feasibility and equity under varying difficulty conditions, though extremely imbalanced difficulty profiles can challenge both solution quality and workload distribution.

Table 7. MRAP runtimes with 10 leagues and 12 teams with harder games to call.

Game distribution Normal,hard,very hard (%)	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
80,15,5 (Baseline)	2.24	0.00	0.00	0.00	72.00	2.80	10.00
60,25,15	2.24	0.00	0.00	0.00	71.80	2.40	10.00
5,15,80	3.01	0.00	343.70	158.51	69.60	0.20	10.00

Table 8 analyzes how tightening specific assignment constraints affects the computational performance of the MRAP model. The baseline case (referred to as "With defaults") assumes moderately relaxed settings: Each referee can officiate up to four games per team, including a maximum of three home games, and must rest at least once every four weeks. Three constraint variations are tested individually: "Max. home games = 1" sets $H^{\max}_{r,t} = 1$ for all $r \in R, t \in T$ in Constraints (11); "Max. for a team = 2" sets $T^{\max}_{r,t} = 2$ for all $r \in R, t \in T$ in Constraints (10); and "Rest every 2 weeks" assumes $MW = 2$ in Constraints (15). The problem size is held constant across all scenarios (10 leagues, 12 teams, 72 referees), allowing for a direct comparison of their effects on runtime, solution quality, and workload distribution. Requiring more frequent rest periods slightly increases runtime, while limiting the number of games per team or per home venue slightly decreases the computational effort. Importantly, Dummy assignments, Games under par, and Total under par penalties remain zero across all scenarios, indicating that the solver consistently finds feasible and high-quality solutions despite the tighter constraints. The minimum number of games assigned per referee increases when team-related constraints are tightened due to reduced assignment flexibility, forcing referees to be utilized more uniformly. Interestingly, in the "Rest every 2 weeks" case, the minimum number of games per referee further increases to 6.40. This behavior likely also reflects a redistribution of workload among the remaining active time slots: Although referees rest more frequently, the need to cover all games compresses assignments into fewer available rounds, leading to less variability in workloads.

Table 8. MRAP runtimes with 10 leagues and 12 teams with different constraints.

Condition	Runtime (secs.)	Dummy assignments	Games under par	Total under par	Referees used	Min. games	Max. games
With defaults	2.24	0.00	0.00	0.00	72.00	2.80	10.00
Max. home games = 1	2.11	0.00	0.00	0.00	72.00	5.00	10.00
Max. for a team = 2	2.10	0.00	0.00	0.00	72.00	4.20	10.00
Rest every 2 weeks	2.54	0.00	0.00	0.00	72.00	6.40	10.00

6. Conclusion

Organizing tournaments in non-professional sports, such as youth leagues, is challenging due to the large number of games involved. One of the critical issues in this process is assigning referees to games across multiple leagues. This paper introduces the *Multi-League Referee Assignment Problem (MRAP)*, a practically motivated scheduling challenge that arises in non-professional football leagues. The MRAP generalizes the classical Referee Assignment Problem by incorporating *multiple parallel leagues* with structured constraints reflecting real-world practices, such as league eligibility, fairness in workload, rest policies, and performance-based matching of referees to game difficulty. We presented a detailed integer programming formulation of the MRAP that captures these constraints comprehensively. While the monolithic model provides optimal solutions and guarantees high-quality assignments, it was shown to be computationally infeasible for realistically sized instances.

To address scalability issues, we proposed two decomposition-based heuristic methods: temporal decomposition by rounds and structural decomposition by leagues. Extensive computational experiments demonstrate that both heuristics achieve substantially faster runtimes than the monolithic formulation while maintaining solution quality, with temporal decomposition (weekly batching) using shorter batches proving particularly effective. Counterintuitively, we found that extending the batch length beyond a single round often degrades solution quality, as longer batches create suboptimal commitments that excessively constrain subsequent subproblems. Additionally, a series of sensitivity analyses revealed how key modeling parameters (such as referee availability, game difficulty distributions, and constraint tightness) affect both assignment feasibility and computational performance. These heuristic methods can easily be adapted to cases where game difficulties and referee scores change dynamically as the season progresses by re-solving the assignment problem in intermediate rounds with updated parameters for the remainder of the season.

There also remain several directions for future research. In practice, referee performance scores and availability are not deterministic. Integrating stochastic programming or robust optimization techniques into the MRAP would enable the model to handle uncertainties more effectively in referee evaluation or last-minute unavailability. This would enhance the model's applicability in dynamic and volatile scheduling environments. Currently, assignments are generated in batch modes. Developing online algorithms that adjust assignments in real time in response to new information (e.g., cancellations, referee withdrawals) would further improve practical usability, particularly in amateur settings with high uncertainty. Although the weekly batching scheme advances over non-overlapping batches rather than as a rolling horizon, a natural extension is to revisit earlier batches by sweeping the optimization window both forward and backward in time until no further improvement is obtained; this iterative re-optimization may recover some of the quality lost to early-batch commitments. Referee performance is often influenced by factors such as game pressure, rivalries, or crowd behavior. Leveraging machine learning techniques to predict performance trends or to inform suitability scores (e.g., for computing mismatch penalties) could enrich the decision-making process. Future models could also integrate the Traveling Umpire Problem considerations, especially relevant in geographically dispersed tournaments.

Regarding real-world implementation, we note that during this research a prototype decision support system was developed. We consider effective implementation to be feasible, and we outline a brief roadmap here. A practical deployment would begin with a data integration step, mapping the federation's existing referee registry, game schedules, and eligibility records onto the model parameters. A calibration phase would follow, using historical assignments to validate that the model's outputs are consistent with expert judgment. The system would then be piloted on a subset of leagues with outputs compared against manually constructed assignments. Feedback from assignment coordinators would inform constraint refinement before a broader rollout. A formal case study comparing optimized and manual assignments in terms of fairness, workload balance, and mismatch penalties remains an important direction for future work. We also note that the sensitivity analyses reported here were conducted under simplified experimental conditions – in particular, with zero minimum workload thresholds and full referee availability – which tend to produce mild effects. Deployments involving non-zero lower bounds or partial referee availability may exhibit greater sensitivity to parameter choices, and we encourage practitioners to conduct instance-specific sensitivity analyses before deployment.

In conclusion, this work lays the foundation for a generalized and scalable framework for referee assignment across multiple leagues. The proposed methods offer a promising balance between optimality and practical implementation. With further enhancements by incorporating stochastic modeling, real-time adaptation, and hybrid learning-integrated frameworks, the MRAP can evolve into a comprehensive decision support tool for sports federations worldwide.

Funding

This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Grant 121M653.

ORCID

Tankut Atan: https://orcid.org/0000-0002-3241-4617
Burak Çavdaroğlu: https://orcid.org/0000-0003-2756-5370
Zühal Özcan: https://orcid.org/0000-0001-5719-0974

References

[1] Alarcón, F., Durán, G., and Guajardo, M. Referee assignment in the Chilean football league using integer programming and patterns. *International Transactions in Operational Research 21*, 3 (2014), 415–438.

[2] Atan, T., and Hüseyinoğlu, O. P. Simultaneous scheduling of football games and referees using Turkish league data. *International Transactions in Operational Research 24*, 3 (2017), 465–484.

[3] Battistutta, M., Bisson, M., De Cesco, F., and Schaerf, A. Referee assignment for a basketball league with multiple divisions using local search. In *Proceedings of the 12th Metaheuristics International Conference* (2017), MIC 2017, pp. 339–341.

[4] Bender, M., and Westphal, S. A combined approximation for the traveling tournament problem and the traveling umpire problem. *Journal of Quantitative Analysis in Sports 12*, 3 (2016), 139–149.

[5] Çavdaroğlu, B., and Atan, T. Determining matchdays in sports league schedules to minimize rest differences. *Operations Research Letters 48*, 3 (2020), 209–216.

[6] Chandrasekharan, R. C., Toffolo, T. A., and Wauters, T. Analysis of a constructive matheuristic for the traveling umpire problem. *Journal of Quantitative Analysis in Sports 15*, 1 (2019), 41–57.

[7] Davari, M., Goossens, D., Beliën, J., Lambers, R., and Spieksma, F. C. The multi-league sports scheduling problem, or how to schedule thousands of matches. *Operations Research Letters 48*, 2 (2020), 180–187.

[8] de Werra, D. Scheduling in sports. In *Studies on Graphs and Discrete Programming* (1981), P. Hansen, Ed., North-Holland, pp. 381–395.

[9] Dinitz, J. H., and Stinson, D. R. On assigning referees to tournament schedules. *Bulletin of the Institute of Combinatorics and its Applications 44* (2005), 22–28.

[10] Duarte, A. R., Ribeiro, C. C., and Urrutia, S. A hybrid ILS heuristic to the referee assignment problem with an embedded MIP strategy. In *International Workshop on Hybrid Metaheuristics* (2007), Springer, pp. 82–95.

[11] Duarte, A. R., Ribeiro, C. C., Urrutia, S., and Haeusler, E. H. Referee assignment in sports leagues. In *International Conference on the Practice and Theory of Automated Timetabling* (2006), Springer, pp. 158–173.

[12] Durán, G., Guajardo, M., and Gutiérrez, F. Efficient referee assignment in Argentinean professional basketball leagues using operations research methods. *Annals of Operations Research* (2021), 1–19.

[13] Durán, G., Guajardo, M., López, A. F., Marenco, J., and Zamorano, G. A. Scheduling multiple sports leagues with travel distance fairness: An application to Argentinean youth football. *INFORMS Journal on Applied Analytics 51*, 2 (2021), 136–149.

[14] Evans, J. R. A microcomputer-based decision support system for scheduling umpires in the American baseball league. *Interfaces 18*, 6 (1988), 42–51.

[15] Evans, J. R., Hebert, J. E., and Deckro, R. F. Play ball!–the scheduling of sports officials. *Perspectives in Computing: Applications in the Academic and Scientific Community 4*, 1 (1984), 18–29.

[16] Farmer, A., Smith, J. S., and Miller, L. T. Scheduling umpire crews for professional tennis tournaments. *Interfaces 37*, 2 (2007), 187–196.

[17] Hawkey, M. *Methodology Informing the Assignment of Match Referees in Professional Soccer*. PhD thesis, Victoria University, Australia, 2023.

[18] Lafuente, J. G. The best systems for appointing referees. In *Economics, management and optimization in sports*. Springer, 2004, pp. 101–120.

[19] Li, M., Davari, M., and Goossens, D. Multi-league sports scheduling with different leagues sizes. *European Journal of Operational Research 307*, 1 (2023), 313–327.

[20] Li, M., and Goossens, D. Grouping and scheduling multiple sports leagues: An integrated approach. *Journal of the Operational Research Society 76*, 4 (2025), 739–757. doi: 10.1080/01605682.2024.2391516.

[21] Linfati, R., Gatica, G., and Escobar, J. W. A flexible mathematical model for the planning and designing of a sporting fixture by considering the assignment of referees. *International Journal of Industrial Engineering Computations 10* (2019), 281–294.

[22] Mancini, S., and Isabello, A. Fair referee assignment for the Italian soccer Serie A. *Journal of Quantitative Analysis in Sports 10*, 2 (2014), 153–160.

[23] Moody, D., Kendall, G., and Bar-Noy, A. Youth sports leagues scheduling. *PATAT 2010* (2010), 283–293.

[24] Samuel, R. D., Galily, Y., and Hochman, G. The "silent" noise: moving forward from bias to noise in football referees' decision-making. *International Journal of Sport and Exercise Psychology 24*, 3 (2026), 405–430.

[25] Scarelli, A., and Narula, S. C. A multicriteria assignment problem. *Journal of Multi-Criteria Decision Analysis 11*, 2 (2002), 65–74.

[26] The Football Association. Rules of the Association 2021-22. https://www.thefa.com/-/media/files/thefaportal/governance-docs/rules-of-the-association/2021-22/referees.ashx, 2021. Accessed: 2025-11-15.

[27] The Football Association. An update on proposed structural changes to the national league system, 2021. Accessed November 15, 2025. https://www.thefa.com/news/2021/apr/12/nls-restructure-20211204.

[28] Toffolo, T. A., Wauters, T., Van Malderen, S., and Berghe, G. V. Branch-and-bound with decomposition-based lower bounds for the traveling umpire problem. *European Journal of Operational Research 250*, 3 (2016), 737–744.

[29] Trick, M. A., and Yildiz, H. Benders' cuts guided large neighborhood search for the traveling umpire problem. *Naval Research Logistics 58*, 8 (2011), 771–781.

[30] Trick, M. A., and Yildiz, H. Locally optimized crossover for the traveling umpire problem. *European Journal of Operational Research 216*, 2 (2012), 286–292.

[31] Trick, M. A., Yildiz, H., and Yunes, T. Scheduling major league baseball umpires and the traveling umpire problem. *Interfaces 42*, 3 (2012), 232–244.

[32] Vallespi Soro, F. The referee assignment problem. B.S. thesis, Universitat Politècnica de Catalunya, 2019.

[33] Xue, L., Luo, Z., and Lim, A. Two exact algorithms for the traveling umpire problem. *European Journal of Operational Research 243*, 3 (2015), 932–943.

[34] Yavuz, M., İnan, U. H., and Fiğlali, A. Fair referee assignments for professional football leagues. *Computers & Operations Research 35*, 9 (2008), 2937–2951.