Can LLMs Assign Match Referees with Reasoning?

Yizhou Xu
KTH Royal Institute of Technology
yizhoux@kth.se

1 Introduction

This project aims to automatically assign referees to football matches based on referee availability, experience, match importance, and fatigue constraints. Our primary focus is to explore whether a reasoning-capable large language model (LLM), such as OpenAI o3 and o4-mini, can effectively perform this type of structured decision-making. As a baseline, we implement the same problem using a symbolic solver (MiniZinc), which could provide an optimal solution under clearly defined rules. This comparison allows us to evaluate the reasoning capabilities of current LLMs against traditional, rule-based optimization systems. Our code and data are available at `https://github.com/xyz961014/RefereeScheduler`.

2 Background

In both professional and amateur football leagues, scheduling weekend matches involves more than just coordinating teams and venues. It also requires assigning qualified referees to each game. Every game must be officiated by a team of four officials:

- one main referee
- two assistant referees
- one fourth official

Scheduling the appropriate referees is a complex logistical task directly impacting match quality and fairness.

Referees differ in their experience levels, have varying availability time windows, and must be managed to avoid fatigue, particularly when they are assigned to multiple matches in close succession. Additionally, matches vary in difficulty and importance. Higher-tier games usually require more experienced referees, while lower-tier games may be suitable for less experienced officials. These factors make the assignment process a balance between fairness and efficiency under real-world constraints.

Traditionally, human specialists make referee assignments manually, often relying on rules of thumb and institutional knowledge. This project explores whether this process can be automated through large language models (LLMs), while a symbolic solver (MiniZinc) is also implemented as a baseline system.

3 Problem Definition

We aim to schedule referees for a **single day** of football matches, ensuring that each game is properly officiated and all assignments respect referee constraints such as availability, fatigue, and experience.

3.1 Entities and Properties

Game

Each game is defined by:

- `time_begin`, `time_end`: the scheduled start and end times (e.g., `12:00-14:00`)
- `difficulty_factor`: a numerical value representing match difficulty, influenced by factors such as league tier, competing teams, and possibly other attributes
- `field`: the physical location where the match takes place

Referee

Each referee is characterized by:

- `available_time_slots`: a set of time intervals during which the referee is available
- `main_ref_experience`: the cumulative difficulty factor of all previously officiated matches as a **Main Referee**
- `assistant_ref_experience`: the cumulative difficulty factor of all previously officiated matches as an **Assistant Referee**

Note: We do not consider the role of Fourth Official to require experience, and it is not considered for fatigue.

3.2 Constraints

1. **No Time Conflicts**
   A referee cannot be assigned to overlapping matches.
2. **Role Fatigue Limit**
   A referee may be assigned to **at most two matches** in the roles of **Main Referee** or **Assistant Referee**. Additional assignments are not allowed due to fatigue constraints.
3. **Travel Time Between Fields**
   If a referee is assigned to consecutive matches at different fields, there must be at least **30 minutes** of travel time between the end of the first match and the start of the next.

3.3 Objective

Maximize the overall assignment quality by prioritizing experienced referees for difficult games:

- Assign **experienced Main Referees** to **high-difficulty matches** as Main Referees
- Assign **experienced Assistant Referees** similarly for Assistant Referee roles

This objective ensures high-quality officiating while maintaining fairness and feasibility across all assignments.

4 Method

We implement two approaches to evaluate automated referee assignment: a symbolic solver and a reasoning-based large language model (LLM) system.

4.1 Symbolic Solver Baseline: MiniZinc Model

As a baseline, we model the referee assignment problem using MiniZinc, a high-level constraint modeling language[1].

[1] `https://www.minizinc.org/`

All constraints described above are implemented within the MiniZinc model, including availability, fatigue, non-overlap, travel buffer, and role exclusivity. Please refer to our repository for the full model and data format.

The objective function is defined to maximize overall assignment quality by favoring experienced referees for more challenging (high-difficulty) matches. The model is solved using the `chuffed` backend solver, with a fixed time limit to bound computational effort.

Although MiniZinc guarantees that all constraints are strictly satisfied, it does not guarantee optimality when a timeout is set. In such cases, the solver returns the best feasible solution found within the time budget (i.e., a valid but potentially suboptimal assignment).

4.2 LLM-Based Assignment with Prompting

The second method leverages large language models (LLMs) that are capable of reasoning. We use **OpenAI o3**, **o4-mini**, and **DeepSeek-R1** to perform the same assignment task through the same structured prompting. In this approach, the full game and referee data are embedded directly into a text-based prompt, along with a clear specification of constraints and expected output format (JSON).

The model output is parsed and validated against hard constraints. Notably, the LLM is not guaranteed to find an optimal solution or even a feasible one, but it serves as a test of whether LLMs can reason effectively over structured scheduling constraints when guided by a natural language specification.

5 Experiment

5.1 Data

The dataset consists of anonymized scheduling records provided by the Tsinghua University Student Football Association, with minor adjustments. It consists of 20 different days of football matches, comprising approximately nine games per day and involving around 25 referees per day. Each entry includes information about match timing, field location, difficulty level, and referee availability and experience. The data is available in the repository.

5.2 Metric

We define the assignment quality using a weighted experience score. The total score is calculated as:

$$\begin{aligned}\text{Score} = \sum_{\text{game}} \texttt{game\_difficulty} \times (&\texttt{main\_referee\_experience} \\ &+\texttt{assistant\_referee\_1\_experience} \\ &+\texttt{assistant\_referee\_2\_experience})\end{aligned} \tag{1}$$

If any constraint is violated, the score is set to zero:

$$\text{Score} = 0 \quad \text{if any constraint is not satisfied}$$

To normalize performance across different days, we also compute the average score per game (SPG):

$$\text{SPG} = \frac{\text{Score}}{\#\text{Games}}$$

5.3 Results

Table 1 summarizes the performance of both symbolic and LLM-based methods across varying time or reasoning budgets. We evaluate results with average score per game (SPG). We also report average time per day (TPD).

For the symbolic baseline using MiniZinc with the `chuffed` solver, we observe that increasing the time budget yields slightly better results. However, even at 120 seconds, the solver does not always

Method	Model	Timeout/Reasoning Effort	SPG	TPD
Symbolic Solver	MiniZinc chuffed	10	322.28	10.23
		60	386.74	58.51
		120	**396.34**	113.72
LLMs	OpenAI o3	medium	100.58	278.15
	OpenAI o4-mini	high	255.74	377.26
	DeepSeek-R1	high	0.00	404.10

Table 1: Comparison of LLM-based methods and symbolic solver. SPG: Score per game. TPD: Time per day (average inference or solve time over a 24-hour schedule).

reach the optimal solution due to the problem's combinatorial complexity. Nevertheless, all solutions generated by MiniZinc strictly satisfy all constraints, ensuring feasibility.

In contrast, LLM-based methods often fail to satisfy hard constraints such as role uniqueness, time conflicts, or fatigue limits. As a result, their SPG scores are significantly lower. DeepSeek-R1, in particular, produces infeasible outputs in all test cases, yielding a score of zero. Among the LLMs evaluated, OpenAI's `gpt-4o-mini` performs the best, achieving a much higher SPG than other models, but still far behind the symbolic solver. Moreover, LLM-based approaches are often slower. Their time per day (TPD) exceeds that of symbolic solvers.

These results indicate that while LLMs can reason about structured data to some extent, they currently lack the reliability and precision required for constrained scheduling problems. Their outputs may resemble correct solutions, but they often violate important feasibility rules, suggesting a potential research area for improving LLMs.

6 Conclusion

This project investigates whether LLMs can perform structured, constraint reasoning in the context of referee assignment for football matches. By comparing LLM-generated solutions against a symbolic solver baseline (MiniZinc), we assess their ability to handle real-world scheduling constraints such as availability, fatigue, role exclusivity, and travel time.

Our findings suggest that current LLMs are not yet capable of reliably solving such constrained assignment tasks. While they can generate syntactically correct and contextually plausible outputs, they frequently violate hard constraints, resulting in infeasible solutions.

These results highlight a current limitation of LLMs in constraint satisfaction and structured planning. Evaluating and advancing LLMs in this direction remains a significant challenge for future research.