CDMO

Cono Cirone

July 2025

1 Introduction

2 CP Model

2.1 Decision Variables

This model utilizes two primary decision variables to construct the tournament schedule and assign home/away teams for each match.

2.1.1 matches

The variable $\text{matches}_{p,w} \in [0, \ldots, \frac{N}{2} - 1]$ for $p \in P, w \in W$, determines which match from the predefined round-robin structure (rb) is scheduled in week w and period p.

2.1.2 home_away

The binary variable $\text{home\_away}_{p,w} \in \{0, 1\}$ for $p \in P, w \in W$, assigns the home team for the match scheduled at period p in week w. Specifically, $\text{home\_away}_{p,w} = 0$ means the first team listed in the match plays at home, while $\text{home\_away}_{p,w} = 1$ means the second team plays at home.

2.2 Auxiliary Variables

2.2.1 home_games

The auxiliary variable $\text{home\_games}_t \in [1, \ldots, N - 1]$ for $t \in T$, represents the total number of home games assigned to team t throughout the tournament. Its value is derived from the assignments of the decision variables matches and home_away.

2.3 Objective Function

The model's objective is to quantify and minimize disparities in home game assignments through the max_imbalance variable.

2.3.1 Objective Variable: max_imbalance

The integer variable max_imbalance $\in [1, \ldots, N-1]$ quantifies the maximum absolute disparity in home game assignments across all teams. The lower bound of 1 acknowledges that perfect balance (max_imbalance = 0) is not achievable for an odd number of total games ($N-1$ games). The upper bound of $N-1$ represents the theoretical maximum possible deviation, occurring if a team plays all its games either at home or away.

The value of max_imbalance is determined by the following fairness constraint:

$$\forall t \in T : |2 \times \text{home\_games}_t - (N-1)| \leq \text{max\_imbalance}$$

This formulation precisely defines the absolute "imbalance" for each team t. This imbalance is derived from the difference between a team's home games and away games. Since $\text{home\_games}_t + \text{away\_games}_t = N-1$ (total games), substituting the expression for away_games_t yields the imbalance for team t as $2 \times \text{home\_games}_t - (N-1)$. By enforcing that the absolute value of this imbalance for every team must be less than or equal to max_imbalance, this variable effectively captures the largest such deviation among all teams, serving as the direct measure of the overall schedule's fairness.

2.3.2 Objective

The objective is to **minimize max_imbalance**:

$$\text{minimize max\_imbalance}$$

This aims to achieve the fairest possible distribution of home and away games. An optimal solution would ideally yield max_imbalance = 1, meaning each team's home/away game count differs by at most one, which is the best possible outcome for an odd number of total games played.

2.4 Constraints

2.4.1 Core Constraints

These constraints are strictly necessary for defining a feasible round-robin schedule:

1. **Each period must be used exactly once per week:** Ensures that for every week, all matches generated by the round-robin structure's periods are indeed scheduled. Without this, some pairings might be missed, or periods might be duplicated, leading to an incomplete or invalid schedule.

$$\forall w \in W : \texttt{all\_different}([\texttt{matches}[p, w] \mid p \in P])$$

2. **Each team plays at most twice per period:**

$$\forall p \in P, \forall t \in T : \left|\{(w, s) \mid w \in W, s \in S, \text{rb}_{\text{matches}_{p,w}, w, s} = t\}\right| \leq 2$$

3. **Calculation of Home Games:** This constraint defines the value of home_games_t.

$$\forall t \in T : \text{home\_games}_t = \sum_{p \in P, w \in W, s \in S} \mathbb{I}\left(\text{rb}_{\text{matches}_{p,w},w,s} = t \wedge \text{home\_away}_{p,w} = s\right)$$

The indicator function $\mathbb{I}(\cdot)$ ensures that 1 is added to the sum if team t is located in slot s and that slot s is designated as the home slot by $\text{home\_away}_{p,w}$.

4. **Fairness - balance home and away games:** This constraint directly links the calculated home_games for each team to the objective variable max_imbalance. For a detailed explanation of this relationship and the definition of max_imbalance, please refer to Section 2.3.1 (Objective Variable and Objective Function).

2.4.2 Implied Constraints

1. **Each team appears exactly once per week:**

$$\forall w \in W, \forall t \in T : \left|\{(p,s) \mid p \in P, s \in S, \text{rb}_{\text{matches}_{p,w},w,s} = t\}\right| = 1$$

2.4.3 Symmetry Breaking Constraints

1. **Break period assignment symmetry using lexicographic ordering:** The order in which matches corresponding to P are assigned within matches for each week is symmetrical. This constraint breaks such symmetries by enforcing a lexicographical ordering, reducing the number of equivalent search paths.

$$(\text{matches}_{p,w})_{p \in P, w \in W} \succeq_{\text{lex}} (\text{matches}_{p,w})_{\text{reversed}(p) \in P, w \in W}$$

This states that the sequence of matches variables, when read in normal (p, w) order, must be lexicographically greater than or equal to when read in (p, w) order with p reversed. This helps to fix one permutation of period assignments.

2. **Fix first match home assignment to break home/away symmetry:** This constraint eliminates global home/away assignment symmetry by fixing the home/away status of the first match.

$$\text{home\_away}_{0,0} = 0$$

3. **Balance the home/away assignments within each week:** While not a strict symmetry breaking constraint, this constraint helps reduce the search space by ensuring that within each week w, the number of

matches where the second team plays at home ($\text{home\_away}_{p,w} = 1$) is roughly half of the total matches ($|P|$), with a maximum deviation of 1:

$$\forall w \in W : \left| \sum_{p \in P} \text{home\_away}_{p,w} - \left\lfloor \frac{|P|}{2} \right\rfloor \right| \leq 1$$

This guides the solver towards balanced assignments and prunes highly imbalanced weekly configurations.

2.5 Validation

The model was implemented in MiniZinc and validated through a series of experiments designed to assess solver performance under various model configurations and search strategies.

2.5.1 Experimental Design

To comprehensively evaluate the performance of different solving strategies for the Sports Tournament Scheduling problem, a systematic experimental study was conducted.

Hardware and Software: Experiments were executed on a MacBook Air M1 equipped with an 8-core CPU. The following solvers were employed: *Gecode*, *Chuffed* and *OR-Tools CP-SAT*. A uniform time limit of 300 seconds was imposed for each individual problem instance.

Model Configurations: Four configurations were tested: `baseline (core)`, `baseline+implied`, `baseline+symmetry breaking`, `full model`.

Search Strategies: Three distinct search strategies were employed to analyze solver behavior, with a particular focus on how they influenced Gecode, often considered to have weaker default heuristics compared to modern SAT-based solvers.

Search Strategies: Three distinct search strategies were employed to analyze solver behavior, focusing on their influence on Gecode, given its often weaker default heuristics compared to modern SAT-based solvers.

1. **Default Search Strategy (Solver's Default):** Each solver relied entirely on its built-in decision heuristics and restart policies, serving as a baseline for their inherent capabilities.

2. **Sequential Custom Search Strategy:** A manually defined sequential search (`seq_search`) was applied, prioritizing `matches` variables with `dom_w_deg` and `home_away` variables with `first_fail`, utilizing a `restart_luby(100)` policy.

3. **Relax-and-Reconstruct (LNS) Strategy:** This higher-level strategy incorporated `relax_and_reconstruct` on the `matches` variables (preserving 60% of solution values), leveraging Large Neighborhood Search (LNS) techniques. It was layered on top of the "Sequential Custom Search Strategy."

Solver-Specific Strategy Application: To ensure a fair and controlled comparison under single-threaded conditions (aligning with project constraints), OR-Tools CP-SAT was run without multi-threading. For both Chuffed and OR-Tools CP-SAT, the `free_search` parameter was explicitly omitted when applying the custom Sequential Custom Search and Relax-and-Reconstruct strategies. This allowed direct evaluation of the user-defined MiniZinc search annotations, rather than the solvers' highly optimized default heuristics.

2.5.2 Experimental Results

n	GECODE				CHUFFED				CP-SAT			
	bs	complete	noIMPL	noSB	bs	complete	noIMPL	noSB	bs	complete	noIMPL	noSB
6	0	0	0	0	0	0	0	0	0	0	0	0
8	6	6	6	6	0	0	0	0	0	0	0	0
10	N/A	N/A	N/A	N/A	0	0	0	0	1	1	1	1
12	N/A	N/A	N/A	N/A	60	6	5	6	4	3	3	3
14	N/A	N/A	N/A	N/A	175	175	180	194	5	5	5	5
16	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	15	15	15	15
18	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	273	265	265	265
20	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	85	84	84	84
22	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	239	226	247	218

Table 1: CPU time in seconds for finding the *optimal solution* using *Default Search Strategy (Solver's Default)*

n	GECODE				CHUFFED				CP-SAT			
	bs	complete	noIMPL	noSB	bs	complete	noIMPL	noSB	bs	complete	noIMPL	noSB
6	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	1	1	1
12	0	0	0	0	0	0	0	0	55	56	58	48
14	4	7	4	7	N/A	53	34	N/A	N/A	N/A	N/A	N/A
16	N/A	N/A	191	164	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 2: CPU time in seconds for finding the *optimal solution* using *Sequential Custom Search Strategy*

n	GECODE				CHUFFED				CP-SAT			
	bs	complete	noIMPL	noSB	bs	complete	noIMPL	noSB	bs	complete	noIMPL	noSB
6	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	1	1	1
12	0	0	0	0	9	1	3	2	56	62	59	52
14	1	5	1	0	N/A	89	187	258	N/A	N/A	N/A	N/A
16	184	19	1	8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
18	N/A	3	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 3: CPU time in seconds for finding the *optimal solution* using *Relax-and-Reconstruct (LNS) Strategy*