A Multi-Paradigm Approach to the Sports Tournament Scheduling Problem

Andrea Dotti, andrea.dotti4@studio.unibo.it
Tommaso Petrelli, tommaso.petrelli2@studio.unibo.it

1 Introduction

This report presents a comprehensive investigation of the Sports Tournament Scheduling (STS) problem [2], focusing on both decision and optimisation variants. Our primary objective is to formulate and implement the STS problem across the CP, SAT, SMT, and MIP computational paradigms and to conduct a rigorous empirical evaluation of their relative performance.

In preparing this work, we studied relevant literature to understand the problem structure and to guide our modelling choices and implementations, including [3], [5], and [6].

The complete source code, experimental data, and supplementary materials for this project are publicly available at our GitHub repository [1].

1.1 Common Formalization

Instance parameter. The primary input is an even integer $n \geq 2$ which indicates the number of participating teams.

Derived sets. From n, the following common parameters are derived:

- **Teams.** A set of teams, $\mathcal{T}$, indexed by $t \in \{1, 2, \ldots, n\}$. The total number of teams is $|\mathcal{T}| = n$.
- **Weeks.** A set of weeks, $\mathcal{W}$,indexed by $w \in \{1, 2, \ldots, n-1\}$. The total number of weeks is $|\mathcal{W}| = n - 1$.
- **Periods.** A set of periods, $\mathcal{P}$, indexed by $p \in \{1, 2, \ldots, n/2\}$. The total number of periods per week is $|\mathcal{P}| = n/2$.

1.2 Hard Constraints

For a solution to be valid, we always require:

1. **Unique matches**: for every unordered pair of distinct teams $\{i, j\}$, there must be exactly one match where team i plays against team j.
2. **One game per week**: every team plays exactly once per week.
3. **At most twice per period**: for each period p and team t, team t appears in period p at most twice over the whole tournament.

1.3 Soft Constraints

To enhance solver performance, our models incorporate additional constraints designed to reduce the search space by eliminating symmetric solutions.

1. **Match canonical orientation**: for every unordered pair of distinct teams $\{i, j\}$, impose $i < j$ to remove symmetric valid matches. For instance, since matches $(1, 2)$ and $(2, 1)$ are equivalent, we remove the possible pair $(2, 1)$.
2. **Fix week 1**: fix the first week $w = 1$ to the canonical matching $(1, 2), (3, 4), \ldots, (n-1, n)$ to break the global rotational / week-permutation symmetry.
3. **Week symmetry breaking**: to eliminate symmetric schedules produced by permutations of whole weeks, we enforce a lexicographic ordering between the assignment vectors of consecutive weeks. Let $\prec_{\text{lex}}$ denote the standard lexicographic order on finite tuples (i.e. $x \prec_{\text{lex}} y$ iff there exists an index i such that $x_j = y_j$ for all $j < i$ and $x_i < y_i$). Thus, we impose:
$$\mathbf{u}_w \;\prec_{\text{lex}}\; \mathbf{u}_{w+1} \qquad \forall\, w \in \{1, \ldots, |\mathcal{W}| - 1\}.$$
4. **Period symmetry breaking**: to eliminate symmetry due to permutations of periods, we enforce a lexicographic ordering between consecutive periods, similarly to what has been done for the weeks. Thus, we impose:
$$\mathbf{v}_p \;\prec_{\text{lex}}\; \mathbf{v}_{p+1} \qquad \forall\, p \in \{1, \ldots, |\mathcal{P}| - 1\}.$$

Such constraints are not essential for correctness but are critical for making larger problem instances tractable.

1.4 Optimization version

We optionally balance home/away assignments. Let H_t and A_t denote the number of home and away games played by team $t \in \mathcal{T}$, respectively. Since each team plays a total of $n - 1$ matches, we have the relation:

$$H_t + A_t = n - 1 \quad \forall t \in \mathcal{T}$$

The imbalance for team t is defined as the absolute difference between its home and away games:

$$I_t = |H_t - A_t| = |H_t - \big((n-1) - H_t\big)| = |2H_t - (n-1)|$$

The objective is to minimise the maximum imbalance observed across all teams:

$$\min \left(\max_{t \in \mathcal{T}} I_t \right)$$

When n is even, $n - 1$ is odd, so the best possible maximum imbalance is 1. In solver encodings, we implement the objective via auxiliary `homeCount` variable and a scalar `maxImbalance`.

1.5 Dual viewpoint: HA vs RR

We propose two modelling strategies:

- **Home–Away (HA)**: variables explicitly assign which team occupies the Home and Away slots of each period/week.
- **Round-Robin (RR)**: weekly unordered pairs are precomputed with the circle method; the solver only permutes which pair goes to which period within its week and whether to swap home/away in that slot for optimisation version.

Circle method. As discussed in [3], for even n, place teams $1, \dots, n-1$ on a circle and fix team n in the center. For round $r \in \{1, \dots, n-1\}$, team n plays team r; among the remaining teams $i, j \in \{1, \dots, n-1\} \setminus \{r\}$, pair i with j iff $i + j \equiv 2r \pmod{n-1}$. This yields $n-1$ disjoint perfect matchings $\{\mathcal{M}_r\}$ (a canonical 1-factorization) ordered cyclically. In our RR models we precompute the unordered pair scheduled in week w and period p as $(A[p, w], B[p, w]) \in \mathcal{M}_w$; the solver then only permutes periods within each week (and, in the optimisation variant, may swap home/away).

1.6 Experimental Setup

The experimental evaluation was conducted using a Dockerized environment consisting of an Ubuntu 24.04.2 LTS (Noble Numbat) Docker container based on the `minizinc/minizinc:latest` image [4]. The host system featured an Intel Core i7-10510U CPU with 16GB of RAM.

We evaluated each model configuration across problem instances with varying sizes $n \in \{4, 6, \dots, 18\}$, imposing a computational timeout of 300 seconds per instance. The experimental results are presented in tabular format, where each row corresponds to a specific instance (indexed incrementally such that ID = 0 represents $n = 4$ and ID = 7 represents $n = 18$). Each cell reports either a single value indicating the execution time in seconds for decision problems, or two values comprising the execution time and the objective value (highlighted in bold) for experiments conducted in both decision and optimisation variants.

All generated solutions underwent verification for correctness using the provided `check_solution.py` validation script to ensure the reliability of the reported results.

2 CP Model

2.1 Decision Variables

HA model.

- $\texttt{Home}[p, w] \in \mathcal{T}$ for every $p \in \mathcal{P}, w \in \mathcal{W}$. Semantics: team playing at home in period p of week w.
- $\texttt{Away}[p, w] \in \mathcal{T}$ analogous for away teams.

RR model.

- $\texttt{pos}[p, w] \in \{1, \dots, |\mathcal{P}|\}$: a permutation variable that selects which precomputed match-index k is placed at slot (p, w). Period/week mapping given by: Home/Away at (p, w) correspond to $\texttt{Home}[\texttt{pos}[p, w], w]$ and $\texttt{Away}[\texttt{pos}[p, w], w]$.
- For optimization: $\texttt{swap}[p, w] \in \{0, 1\}$ toggles orientation of the pair assigned to (p, w).

2.2 Objective Function

The RR optimisation model introduces the auxiliary variable:

$$\texttt{homeCount}[t] = \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}} \Big(\mathbb{I}(\texttt{Home}[\texttt{pos}[p, w], w] = t) \cdot (1 - \texttt{swap}[p, w]) + \mathbb{I}(\texttt{Away}[\texttt{pos}[p, w], w] = t) \cdot \texttt{swap}[p, w] \Big),$$

and the objective variable:

$$\texttt{maxImbalance} = \max_{t \in \mathcal{T}} |2 \cdot \texttt{homeCount}[t] - (n-1)|.$$

The solver minimises `maxImbalance` when the optimisation variant is executed.

2.3 Constraints

2.3.1 Hard constraints

1. **Unique Matches.**

   HA model. Map each match to a canonical integer identifier. This method ensures that a match between teams i and j results in the same unique ID, regardless of whether the home-away assignment is (i, j) or (j, i). Let `matchId` be the canonical identifier linearisation function:

   $$\mathrm{id}(x, y) = \begin{cases} x \cdot n + y & \text{if } x < y \\ y \cdot n + x & \text{if otherwise} \end{cases}$$

   For every match slot in the schedule, we can enforce

   $$\texttt{matchId}[p, w] = \mathrm{id}(\texttt{Home}[p, w], \texttt{Away}[p, w])$$

   and then mandates that all the match IDs are different:

   $$\texttt{alldifferent}(\{\texttt{matchId}[p, w]\}_{p,w}).$$

   RR model The circle method described in Section 1.5 ensures this property by design.

2. **One game per week.**

   HA model. For each week $w \in \mathcal{W}$:

   $$\texttt{alldifferent}(\{\texttt{Home}[p, w] \mid p \in \mathcal{P}\} \cup \{\texttt{Away}[p, w] \mid p \in \mathcal{P}\}).$$

   RR model. Assign each pre-computed match in $\mathcal{M}_w$ from Section 1.5 to a unique period within the week. Thus, for each week $w \in \mathcal{W}$:

   $$\texttt{alldifferent}(\{\texttt{pos}[p, w] \mid p \in \mathcal{P}\}).$$

3. **At most twice per period.**

   HA model. For every period $p \in \mathcal{P}$ and team $t \in \mathcal{T}$:

   $$\sum_{w \in \mathcal{W}} \Big(\mathbb{I}(\texttt{Home}[p, w] = t) + \mathbb{I}(\texttt{Away}[p, w] = t)\Big) \leq 2$$

   RR model. Analogously, by replacing p with $\texttt{pos}[p, w]$.

2.3.2 Soft constraints

1. **Match canonical orientation.** For each match impose $\texttt{Home}[p,w] < \texttt{Away}[p,w]$ to remove symmetric matches.

2. **Fix week 1.**

   HA model. For every period $p \in \mathcal{P}$:

$$\texttt{Home}[p,1] = 2p-1, \ \texttt{Away}[p,1] = 2p.$$

   RR model. Analogously, for every period $p \in \mathcal{P}$ we fix the permutations for week 1 to identity:

$$\texttt{pos}[p,1] = p.$$

3. **Week symmetry breaking.**

   HA model. For each $w \in \{1,\dots,|\mathcal{W}|-1\}$ define the week vector as the concatenation of the Home-block and the Away-block:

$$\mathbf{u}_w \ = \ \Big(\texttt{Home}[p,w]\Big)_{p=1}^{|\mathcal{P}|} \parallel \Big(\texttt{Away}[p,w]\Big)_{p=1}^{|\mathcal{P}|}.$$

   RR model. Analogously, by replacing p with $\texttt{pos}[p,w]$.

4. **Period symmetry breaking.**

   HA model. For each $p \in \{1,\dots,|\mathcal{P}|-1\}$ define the period vector by concatenating the `Home` and `Away` entries across all weeks:

$$\mathbf{v}_p \ = \ \Big(\texttt{Home}[p,w]\Big)_{w=1}^{|\mathcal{W}|} \parallel \Big(\texttt{Away}[p,w]\Big)_{w=1}^{|\mathcal{W}|},$$

   RR model. As above, but replacing p with $\texttt{pos}[p,w]$.

2.4 Validation

2.4.1 Experimental design

We tested HA and RR models using MiniZinc v2.9.2. Solvers: Gecode v6.3.0 and Chuffed v0.13.2. Search strategies:

- `first_fail` with `indomain_min` (both solvers).
- `first_fail` with `indomain_random` and `restart_luby` (Gecode only).

Configurations include with/without symmetry breaking and restart variants. In addition, both decision and optimality have been tested for the RR model.

2.4.2 Experimental results

Table 1 reports HA decision-model runs. Table 2 reports RR results for both decision and optimisation variants.

ID	Gecode + SB	Gecode w/o SB	Chuffed + SB	Chuffed w/o SB	Gecode + SB + RST	Gecode w/o SB + RST
0	UNSAT	UNSAT	UNSAT	UNSAT	UNSAT	UNSAT
1	0	0	0	1	0	0
2	0	N/A	0	N/A	0	1
3	0	N/A	0	N/A	0	N/A
4	0	N/A	4	N/A	8	N/A
5	294	N/A	14	N/A	N/A	N/A

Table 1: Results for HA model with/without symmetry breaking in decision version using Gecode with/without restart strategy and Chuffed without restart strategy.

ID	Gecode + SB	Gecode w/o SB	Chuffed + SB	Chuffed w/o SB	Gecode + SB + RST	Gecode w/o SB + RST
0	UNSAT	UNSAT	UNSAT	UNSAT	UNSAT	UNSAT
1	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**
2	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**
3	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**
4	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**	0 — **1**
5	1 — **1**	0 — **1**	117 — **1**	33 — **1**	1 — **1**	1 — **1**
6	2 — **1**	21 — **1**	N/A — N/A	N/A — N/A	45 — **1**	36 — N/A
7	N/A — N/A	N/A — N/A	N/A — N/A	N/A — N/A	106 — N/A	N/A — N/A

Table 2: Results for RR model with/without symmetry breaking in both decision and optimisation versions Gecode with/without restart strategy and Chuffed without restart strategy.

3 SAT Model

3.1 Decision Variables

For both models, one-hot encoding is used to define the following decision variables.

HA model. For each period p, week w, and team t we define literals:

- $\texttt{Home}[p, w, t] \in \{0, 1\}$, true iff team t plays at home in match slot (p, w).
- $\texttt{Away}[p, w, t] \in \{0, 1\}$ analogous for away teams.

RR model. For each period p, week w, and match-index $k \in \{1, \dots, |\mathcal{P}|\}$ we define literals:

- $\texttt{pos}[p, w, k] \in \{0, 1\}$, true iff match-index k is assigned to match slot (p, w).
- For optimisation: $\texttt{swap}[p, w] \in \{0, 1\}$, a boolean which flips orientation at the chosen match slot.

3.2 Objective Function

In the RR model optimization version, for each team $t \in \mathcal{T}$ and week $w \in \mathcal{W}$, define the weekly home literal

$$H_{t,w} \equiv \bigvee_{p \in \mathcal{P}} \Big(\Big(\bigvee_{k:\, \texttt{Home}[k,w]=t} (\texttt{pos}[p, w, k] \land \neg \texttt{swap}[p, w]) \Big) \lor \Big(\bigvee_{k:\, \texttt{Away}[k,w]=t} (\texttt{pos}[p, w, k] \land \texttt{swap}[p, w]) \Big) \Big).$$

Thus $H_{t,w}$ is true iff t plays at home in week w given the placement `pos` and the flip `swap`.

We minimise the maximum home/away imbalance by an outer binary search on a bound B. For each candidate B, every team's number of home weeks must lie in the allowable interval around $W/2$ implied by B; we enforce this *purely with cardinalities*:

$$\texttt{at\_most\_k}\big(\{H_{t,w}\}_{w\in\mathcal{W}},\, U(B)\big), \qquad \texttt{at\_most\_k}\big(\{\neg H_{t,w}\}_{w\in\mathcal{W}},\, W-L(B)\big).$$

If satisfiable at B, we tighten to a smaller bound; otherwise we relax it. The optimum is the least feasible B (for even n, $B^\star = 1$). The operator `at_most_k` is encoded via sequential counters with standard auxiliary literals.

3.3 Constraints

All constraints are Boolean and use one-hot encodings. We employ the standard primitives `at_least_one` (ALO), `at_most_one` (AMO), `exactly_one` (EO=ALO∧AMO), and `at_most_k` (sequential counters). Symmetry breaking compares sequences of one-hot vectors using the strict lex operator `lex_less_onehot_seq`, which implements "X is lexicographically smaller than Y" via a disjunction of prefix-equality and one-hot strict-comparison clauses.

3.3.1 Hard constraints

1. **Unique matches.**

   HA. Per slot (p, w) select exactly one home and one away team and forbid mirroring:

   $$\texttt{exactly\_one}(\{\texttt{Home}[p,w,t]\}_{t\in\mathcal{T}}), \quad \texttt{exactly\_one}(\{\texttt{Away}[p,w,t]\}_{t\in\mathcal{T}}), \quad \forall t:\ \neg(\texttt{Home}[p,w,t]\wedge\texttt{Away}[p,w,t]),$$

   and each unordered pair appears exactly once over the season:

   $$\forall\, i<j:\ \texttt{exactly\_one}\Big(\big\{(\texttt{Home}[p,w,i]\wedge\texttt{Away}[p,w,j]) \vee (\texttt{Home}[p,w,j]\wedge\texttt{Away}[p,w,i])\big\}_{(p,w)}\Big).$$

   RR. Per slot (p, w) choose exactly one precomputed pair index:

   $$\texttt{exactly\_one}(\{\texttt{pos}[p,w,k]\}_{k=1}^{P}),$$

   and uniqueness of unordered pairs across the season is guaranteed by the circle-method precomputation (Section 1.5) together with the weekly permutation.

2. **One game per week.**

   HA. For every team and week, the team appears exactly once (either home or away):

   $$\forall t\in\mathcal{T},\ \forall w\in\mathcal{W}:\ \texttt{exactly\_one}\big(\{\texttt{Home}[p,w,t]\}_{p\in\mathcal{P}}\ \cup\ \{\texttt{Away}[p,w,t]\}_{p\in\mathcal{P}}\big).$$

   RR. Within each week w, positions form a permutation of $\{1,\ldots,P\}$:

   $$\forall w\in\mathcal{W},\ \forall k\in\{1,\ldots,P\}:\ \texttt{exactly\_one}\big(\{\texttt{pos}[p,w,k]\}_{p\in\mathcal{P}}\big),$$

   which, combined with the circle method, implies every team plays exactly once in week w.

3. **At most twice per period.**

HA. For every $p \in \mathcal{P}$ and $t \in \mathcal{T}$:

$$\texttt{at\_most\_k}\big(\{\texttt{Home}[p,w,t]\}_{w\in\mathcal{W}} \;\cup\; \{\texttt{Away}[p,w,t]\}_{w\in\mathcal{W}},\; 2\big).$$

RR. For fixed p, t, define one literal per week

$$X_{t,w,p} \;\equiv \bigvee_{k:\,\texttt{Home}[w,k]=t \text{ or } \texttt{Away}[w,k]=t} \texttt{pos}[p,w,k],$$

and enforce $\texttt{at\_most\_k}(\{X_{t,w,p}\}_{w\in\mathcal{W}},\, 2)$.

3.3.2 Soft constraints

1. **Canonical orientation (HA).** For each slot (p, w) enforce $\texttt{Home} < \texttt{Away}$ by forbidding $\texttt{Home} = j$ with $j \le i$ when $\texttt{Away} = i$:

$$\forall i \in \mathcal{T}\ \forall j \in \mathcal{T} \text{ with } j \le i :\ \neg\big(\texttt{Home}[p,w,i] \wedge \texttt{Away}[p,w,j]\big).$$

2. **Fix week 1.**

HA. For every $p \in \mathcal{P}$: set $\texttt{Home}[p,1,\,2p-1] = \text{true}$, $\texttt{Away}[p,1,\,2p] = \text{true}$, and all other literals in week 1 to false.

RR. For every $p \in \mathcal{P}$: $\texttt{pos}[p,1,p] = \text{true}$ and $\texttt{pos}[p,1,k] = \text{false}$ for $k \neq p$; in the optimisation variant also fix $\texttt{swap}[p,1] = \text{false}$.

3. **Lexicographic symmetry breaking (weeks and periods).**

HA. Compare consecutive periods p and $p+1$ by applying `lex_less_onehot_seq` to the sequences

$$\big(\{\texttt{Away}[p,w,\cdot]\}_{w\in\mathcal{W}} \parallel \{\texttt{Home}[p,w,\cdot]\}_{w\in\mathcal{W}}\big) \prec_{\text{lex}} \big(\{\texttt{Away}[p+1,w,\cdot]\}_{w\in\mathcal{W}} \parallel \{\texttt{Home}[p+1,w,\cdot]\}_{w\in\mathcal{W}}\big),$$

and analogously for consecutive weeks w and $w+1$ using the concatenation over periods.

RR. Compare consecutive periods and weeks by applying `lex_less_onehot_seq` *directly* to the sequences of team one-hot vectors induced by `pos` and the precomputed tables $\texttt{Away}, \texttt{Home}$. For periods p vs. $p+1$ define

$$\mathbf{s}_p \;=\; \Big(\langle \bigvee_{k:\,\texttt{Away}[k,w]=t} \texttt{pos}[p,w,k]\rangle_{t\in\mathcal{T}} \parallel \langle \bigvee_{k:\,\texttt{Home}[k,w]=t} \texttt{pos}[p,w,k]\rangle_{t\in\mathcal{T}}\Big)_{w\in\mathcal{W}}$$

and post $\texttt{lex\_less\_onehot\_seq}(\mathbf{s}_p, \mathbf{s}_{p+1})$. For weeks w vs. $w+1$ define

$$\mathbf{r}_w \;=\; \Big(\langle \bigvee_{k:\,\texttt{Away}[k,w]=t} \texttt{pos}[p,w,k]\rangle_{t\in\mathcal{T}} \parallel \langle \bigvee_{k:\,\texttt{Home}[k,w]=t} \texttt{pos}[p,w,k]\rangle_{t\in\mathcal{T}}\Big)_{p\in\mathcal{P}}$$

and post $\texttt{lex\_less\_onehot\_seq}(\mathbf{r}_w, \mathbf{r}_{w+1})$.

3.4 Validation

3.4.1 Experimental design

We tested HA and RR models implemented using Z3 Python library and then converted into DIMACS CNF format. Solvers: Z3 v4.15.2 for RR optimality variant, MiniSAT and Glucose for decision variant via PySAT v1.8.dev17 Python library.

Configurations include with/without symmetry breaking. In addition, both decision and optimality have been tested for the RR model.

3.4.2 Experimental results

Table 3 reports HA decision-model runs. Table 4 reports RR results for both decision and optimisation variants. The RR execution times shown here were obtained using Z3's optimisation solver, which differs fundamentally from the satisfiability-only approach used by Glucose and MiniSAT, making direct performance comparisons inappropriate.

ID	MiniSAT + SB	MiniSAT w/o SB	Glucose + SB	Glucose w/o SB
0	UNSAT	UNSAT	UNSAT	UNSAT
1	1	0	1	0
2	3	4	11	4
3	27	18	20	20
4	65	69	138	N/A

Table 3: Results for HA model with/without symmetry breaking using MiniSAT and Glucose in decision version.

ID	MiniSAT + SB	MiniSAT w/o SB	Glucose + SB	Glucose w/o SB	Z3 + SB	Z3 w/o SB
0	UNSAT	UNSAT	UNSAT	UNSAT	UNSAT	UNSAT
1	0	0	0	0	0 — **1**	0 — **1**
2	0	0	0	0	0 — **1**	0 — **1**
3	2	0	2	0	1 — **1**	1 — **1**
4	5	1	5	1	4 — **1**	4 — **1**
5	10	5	17	3	9 — **1**	19 — **1**
6	90	4	N/A	176	50 — **1**	88 — **1**
7	N/A	157	N/A	N/A	152 — **1**	259 — **1**

Table 4: Results for RR model with/without symmetry breaking using MiniSAT and Glucose in decision variant and Z3 in optimisation variant.

4 SMT Model

4.1 Decision Variables

All variables are defined over the integer sort using the theory of linear integer arithmetic (LIA).

HA model. For each period p and week w:

- `Home`$[p, w]$: `Int` with domain $\mathcal{T}$, represents the team playing at home in period p of week w.
- `Away`$[p, w]$: `Int` with domain $\mathcal{T}$, analogous for away teams.

RR model. For each period p and week w:

- $\texttt{pos}[p, w]$: Int with domain $1 \dots P$, represents the index k of the match assigned to slot (p, w).
- For optimisation: $\texttt{swap}[p, w]$: Bool which if true swap the team in $\texttt{Away}[p, w]$ at home and the team in $\texttt{Home}[p, w]$ away.

4.2 Objective Function

In the RR optimisation model, we work in LIA with integer variables. Let $W = |\mathcal{W}| = n - 1$ and obtain the precomputed weekly pairs $\texttt{Home}[w, k], \texttt{Away}[w, k] \in \mathcal{T}$ from the circle method (canonicalised so that $\texttt{Home}[w, k] > \texttt{Away}[w, k]$). We use integer variables $\texttt{pos}[p, w] \in \{1, \dots, P\}$ and, only in optimisation, booleans $\texttt{swap}[p, w]$. The number of home weeks for team t is

$$\texttt{homeCount}[t] \;=\; \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}} \Big(\mathbb{I}\big(\texttt{Home}[w, \texttt{pos}[p, w]] = t \wedge \neg \texttt{swap}[p, w]\big) + \mathbb{I}\big(\texttt{Away}[w, \texttt{pos}[p, w]] = t \wedge \texttt{swap}[p, w]\big) \Big),$$

and we introduce an integer $\texttt{maxImbalance}$ satisfying

$$\texttt{maxImbalance} \;\geq\; \big| 2 \cdot \texttt{homeCount}[t] - W \big| \qquad \forall\, t \in \mathcal{T}.$$

4.3 Constraints

All constraints are expressed in the theory of linear integer arithmetic with `Distinct`, bounded domains, and `ite`/indicator encodings.

4.3.1 Hard constraints

1. **Set domains boundaries.**

   HA. Variable domains: $1 \leq \texttt{Home}[p, w], \texttt{Away}[p, w] \leq n$ for all (p, w).

   RR. For each $w \in \mathcal{W}$, the positions form a permutation:

   $$1 \leq \texttt{pos}[p, w] \leq P \quad \forall p.$$

2. **Unique matches.**

   HA. Linearise unordered pairs by $c_{p,w} = \min(\texttt{Home}[p, w], \texttt{Away}[p, w]) \cdot n + \max(\texttt{Home}[p, w], \texttt{Away}[p, w])$ and post

   $$\texttt{Distinct}\big(\{c_{p,w}\}_{p \in \mathcal{P},\, w \in \mathcal{W}}\big).$$

   RR. Guaranteed by the circle method together with the per-week permutation above.

3. **One game per week.**

   HA. For each $w \in \mathcal{W}$,

   $$\texttt{Distinct}\big(\{\texttt{Home}[p, w]\}_{p \in \mathcal{P}} \;\cup\; \{\texttt{Away}[p, w]\}_{p \in \mathcal{P}}\big).$$

RR. For each $w \in \mathcal{W}$,

$$\texttt{Distinct}\big(\texttt{pos}[1,w],\ldots,\texttt{pos}[P,w]\big).$$

4. **At most twice per period.**

HA. For every $p \in \mathcal{P}$ and $t \in \mathcal{T}$,

$$\sum_{w\in\mathcal{W}} \Big(\mathbb{I}(\texttt{Home}[p,w]=t)+\mathbb{I}(\texttt{Away}[p,w]=t)\Big) \;\leq\; 2.$$

RR. For every $p \in \mathcal{P}$ and $t \in \mathcal{T}$,

$$\sum_{w\in\mathcal{W}} \Big(\mathbb{I}(\texttt{Home}[\texttt{pos}[p,w],w]=t)+\mathbb{I}(\texttt{Away}[\texttt{pos}[p,w],w]=t)\Big) \;\leq\; 2.$$

4.3.2 Soft constraints

1. **Match canonical orientation (HA).** For all (p,w): $\texttt{Home}[p,w] < \texttt{Away}[p,w]$.
2. **Fix week 1.**

HA. For every $p \in \mathcal{P}$: $\texttt{Home}[p,1] = 2p{-}1$ and $\texttt{Away}[p,1] = 2p$.

RR. For every $p \in \mathcal{P}$: $\texttt{pos}[p,1] = p$; in the optimisation variant also fix $\texttt{swap}[p,1] = \textsf{false}$.

3. **Lexicographic symmetry breaking (weeks and periods).**

HA. For consecutive periods p and $p{+}1$:

$$\texttt{lex\_less\_seq}\Big(\ (\texttt{Away}[p,w])_{w=1}^{W} \parallel (\texttt{Home}[p,w])_{w=1}^{W}\ ,\ (\texttt{Away}[p{+}1,w])_{w=1}^{W} \parallel (\texttt{Home}[p{+}1,w])_{w=1}^{W}\ \Big),$$

and analogously for consecutive weeks w and $w{+}1$ using concatenation over periods.

RR. Use the team numbers read at the selected indices: for periods,

$$\mathbf{s}_p = (\texttt{Away}[\texttt{pos}[p,w],w])_{w=1}^{W} \parallel (\texttt{Home}[\texttt{pos}[p,w],w])_{w=1}^{W}\ , \quad \texttt{lex\_less\_seq}(\mathbf{s}_p,\mathbf{s}_{p+1}),$$

and for weeks,

$$\mathbf{r}_w = (\texttt{Away}[\texttt{pos}[p,w],w])_{p=1}^{P} \parallel (\texttt{Home}[\texttt{pos}[p,w],w])_{p=1}^{P}\ , \quad \texttt{lex\_less\_seq}(\mathbf{r}_w,\mathbf{r}_{w+1}).$$

4.4 Validation

4.4.1 Experimental design

Experiments have been conducted for HA and RR models implemented using Z3 Python library and then converted into SMT-LIB format to enable solver comparison. Solvers: Z3 v4.15.2 and CVC5 v1.3.0.

Configurations include with/without symmetry breaking. In addition, both decision and optimality have been tested for the RR model. Nevertheless, tests about optimality have been run only using Z3 solver.

4.4.2 Experimental results

Table 5 reports HA decision-model runs. Table 6 reports RR results for both decision and optimisation variants.

ID	Z3 + SB	Z3 w/o SB	CVC5 + SB	CVC5 w/o SB
0	UNSAT	UNSAT	UNSAT	UNSAT
1	0	0	1	1
2	8	27	48	23

Table 5: Results for HA model with/without symmetry breaking using Z3 and CVC5 in decision version.

ID	Z3 + SB	Z3 w/o SB	CVC5 + SB	CVC5 w/o SB
0	UNSAT	UNSAT	UNSAT	UNSAT
1	0 — **1**	0 — **1**	0	0
2	0 — **1**	0 — **1**	4	1
3	7 — **1**	0 — **1**	45	15
4	14 — N/A	5 — N/A	N/A	N/A
5	N/A — N/A	69 — N/A	N/A	N/A

Table 6: Results for RR model with/without symmetry breaking using Z3 and CVC5 in decision variant and Z3 in optimisation variant.

5 MIP Model

5.1 Decision Variables

HA model.

- $y[i, j, p, w] \in \{0, 1\}$, binary decision variable equal to 1 iff the ordered pair of teams (i, j) is scheduled in week $w \in \mathcal{W}$, period $p \in \mathcal{P}$.

For convenience in expressing constraints, we introduce the following auxiliary variables:

- $\texttt{Home}[p, w] = \sum_{(i,j)\in\mathcal{O}} i \cdot y[i, j, p, w] \in \mathcal{T}$: integer variable denoting the home team in slot (p, w), where $\mathcal{O}$ is the set of ordered pairs.
- $\texttt{Away}[p, w] = \sum_{(i,j)\in\mathcal{O}} j \cdot y[i, j, p, w] \in \mathcal{T}$: integer variable denoting the away team in slot (p, w).

RR model.

- $\texttt{pos}[p, w, k] \in \{0, 1\}$, binary decision variable equal to 1 iff precomputed match index $k \in \mathcal{K} = \{1, \ldots, \mathcal{P} + 1\}$ is placed in week $w \in \mathcal{W}$, period $p \in \mathcal{P}$.
- In the optimisation variant: $\texttt{swap}[p, w] \in \{0, 1\}$, binary variable indicating whether the home/away assignment in slot (p, w) is swapped.

5.2 Objective Function

The RR optimisation variant seeks to minimise the maximum imbalance between home and away games across all teams. Let $W = |\mathcal{W}|$ denote the total number of weeks, and let $H_{w,k}$ and $A_{w,k}$ denote the precomputed home and away teams for match index k in week w.

We introduce the following auxiliary binary variables for each slot (p, w) and match index k:

- $y^{\text{home}}_{p,w,k} = 1$ if match k is scheduled in (p, w) and not swapped, and 0 otherwise.
- $y^{\text{away}}_{p,w,k} = 1$ if match k is scheduled in (p, w) and swapped, and 0 otherwise.

These are defined by the following linear constraints:

$$y^{\text{home}}_{p,w,k} \le \texttt{pos}[p, w, k], \quad y^{\text{home}}_{p,w,k} \le 1 - \texttt{swap}[p, w], \quad y^{\text{home}}_{p,w,k} \ge \texttt{pos}[p, w, k] - \texttt{swap}[p, w],$$

$$y^{\text{away}}_{p,w,k} \le \texttt{pos}[p, w, k], \quad y^{\text{away}}_{p,w,k} \le \texttt{swap}[p, w], \quad y^{\text{away}}_{p,w,k} \ge \texttt{pos}[p, w, k] + \texttt{swap}[p, w] - 1.$$

For each team $t \in \mathcal{T}$, we introduce the integer variable $\texttt{homeCount}[t]$ representing the number of home games played by team t:

$$\texttt{homeCount}[t] = \sum_{w\in\mathcal{W},\, p\in\mathcal{P},\, k\in\mathcal{K}} \Big(\mathbb{I}(H_{w,k} = t) \cdot y^{\text{home}}_{p,w,k} + \mathbb{I}(A_{w,k} = t) \cdot y^{\text{away}}_{p,w,k}\Big).$$

Let $\texttt{maxImbalance}$ be an integer variable representing the maximum deviation from a balanced schedule. The absolute value is linearised as:

$$\texttt{maxImbalance} \ge 2\cdot\texttt{homeCount}[t]-W, \quad \texttt{maxImbalance} \ge W-2\cdot\texttt{homeCount}[t], \quad \forall t \in \mathcal{T}.$$

The optimisation problem is then:

$$\min\ \texttt{maxImbalance}.$$

This formulation ensures $\texttt{maxImbalance} \ge \max_{t\in\mathcal{T}} |2 \cdot \texttt{homeCount}[t] - W|$, and the minimisation drives it to the smallest feasible value.

5.3 Constraints

5.3.1 Hard constraints

1. **Unique matches.**

   HA model. Per slot, select exactly one ordered pair:

$$\sum_{(i,j)\in\mathcal{O}} y[i, j, p, w] = 1 \qquad \forall\, w \in \mathcal{W},\ p \in \mathcal{P}.$$

   Across the whole season, each unordered pair appears exactly once. With symmetry breaking:

$$\sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}} y[i, j, p, w] = 1 \qquad \forall\, (i, j) \in \mathcal{O}.$$

   Without symmetry breaking:

$$\sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}} \big(y[i, j, p, w] + y[j, i, p, w]\big) = 1 \qquad \forall\, i < j.$$

RR model. Guaranteed by the precomputed match index $\mathcal{K}$ from the circle method combined with the per-week permutation constraint. In addition we impose per slot, exactly one precomputed pair index:

$$\sum_{k\in\mathcal{K}} \texttt{pos}[p,w,k] = 1 \qquad \forall\, w\in\mathcal{W},\ p\in\mathcal{P}.$$

2. **One game per week.**

HA model.

$$\sum_{p\in\mathcal{P}} \sum_{\substack{(i,j)\in\mathcal{O}\\ t\in\{i,j\}}} y[i,j,p,w] = 1 \qquad \forall\, t\in\mathcal{T},\ w\in\mathcal{W}.$$

RR model. Each week is a permutation of $\mathcal{K}$:

$$\sum_{p\in\mathcal{P}} \texttt{pos}[p,w,k] = 1 \qquad \forall\, w\in\mathcal{W},\ k\in\mathcal{K}.$$

Together with the circle method, this implies each team plays exactly once per week.

3. **At most twice per period.**

HA model.

$$\sum_{w\in\mathcal{W}} \sum_{\substack{(i,j)\in\mathcal{O}\\ t\in\{i,j\}}} y[i,j,p,w] \le 2 \qquad \forall\, t\in\mathcal{T},\ p\in\mathcal{P}.$$

RR model.

$$\sum_{w\in\mathcal{W}} \sum_{\substack{k\in\mathcal{K}\\ t\in\{H_{w,k},A_{w,k}\}}} \texttt{pos}[p,w,k] \le 2 \qquad \forall\, t\in\mathcal{T},\ p\in\mathcal{P}.$$

5.3.2 Soft constraints

1. **Match canonical orientation.**

HA model.

$$\texttt{Home}[p,w] < \texttt{Away}[p,w] \qquad \forall\, p\in\mathcal{P},\ w\in\mathcal{W}.$$

2. **Fix week 1.**

HA model.

$$y[\,2p{-}1,\ 2p,\ p,\ 1\,] = 1 \qquad \forall\, p\in\mathcal{P}.$$

RR model.

$$\texttt{pos}[p,1,k] = \begin{cases} 1 & \text{if } k=p,\\ 0 & \text{if } k\neq p, \end{cases} \qquad \forall\, p\in\mathcal{P},\ k\in\mathcal{K}.$$

In the optimisation variant:

$$\texttt{swap}[p,1] = 0 \qquad \forall\, p\in\mathcal{P}.$$

5.4 Validation

5.4.1 Experimental design

We tested HA and RR MIP models using PuLP v3.2.1 Python library for linear and mixed integer programming with a solver-independent interface. Solvers: GLPK v5.0 and HiGHS v1.7.0.

Configurations include with/without symmetry breaking and restart variants. In addition, both decision and optimality have been tested for the RR model.

5.4.2 Experimental results

Table 7 reports HA decision-model runs. Table 8 reports RR decision and optimisation runs.

ID	GLPK + SB	GLPK w/o SB	HiGHS + SB	HiGHS w/o SB
0	UNSAT	UNSAT	UNSAT	UNSAT
1	0	0	0	0
2	0	N/A	3	6
3	N/A	N/A	28	49

Table 7: Results for HA model with/without symmetry breaking using GLPK and HiGHS.

ID	GLPK + SB	GLPK w/o SB	HiGHS + SB	HiGHS w/o SB
0	UNSAT	UNSAT	UNSAT	UNSAT
1	0 — **1**	0 — **1**	0 — **1**	0 — **1**
2	0 — **1**	0 — **1**	0 — **1**	0 — **1**
3	0 — **1**	0 — **1**	1 — **1**	5 — **1**
4	15 — N/A	24 — N/A	6 — **1**	5 — **1**
5	N/A — N/A	N/A — N/A	30 — **1**	6 — **1**
6	N/A — N/A	N/A — N/A	101 — N/A	11 — N/A

Table 8: Results for RR model in decision and optimisation versions with/without symmetry breaking using GLPK and HiGHS.

6 Conclusions

Our investigation into the Sports Tournament Scheduling problem across multiple paradigms confirms that the Round-Robin (RR) modelling strategy is significantly more efficient and scalable than the Home-Away (HA) approach. The use of precomputed match pairings in the RR model effectively prunes the search space, enabling solutions for larger instances that are intractable for the HA model. While the application of symmetry-breaking constraints is often crucial for solving larger problem instances, our results show that in certain cases, particularly with specific solver-model combinations, models without these constraints can run faster. This suggests a trade-off between the overhead of enforcing symmetry-breaking and the reduction in search space it provides.

Authenticity and Author Contribution Statement

We affirm that the work presented in this report is original and has not been copied or reproduced from external sources without proper citation. All external materials and references have been acknowledged in the bibliography.

AI tools were used solely for minor language refinement and not for content generation or model development.

Both authors contributed equally to the conceptualisation, implementation, and evaluation of the models. All formulations were jointly peer-reviewed, and experimental testing was conducted collaboratively.

References

[1] A. Dotti and T. Petrelli. *sports-tournament-scheduling.* URL: `https://github.com/petrello/sports-tournament-scheduling`.

[2] Z. Kiziltan and R. Amadini. *Project Work Combinatorial Decision Making & Optimization 2024/2025.* University of Bologna, 2025.

[3] Erik Lambrechts et al. "Round-Robin Tournaments Generated by the Circle Method Have Maximum Carry-Over". In: *Integer Programming and Combinatorial Optimization.* Ed. by Quentin Louveaux and Martin Skutella. Cham: Springer International Publishing, 2016, pp. 178–189. ISBN: 978-3-319-33461-5.

[4] MiniZinc. *The official docker images of the MiniZinc bundle.* URL: `https://hub.docker.com/r/minizinc/minizinc`. (accessed: 01.09.2025).

[5] Andrea Schaerf. "Scheduling Sport Tournaments using Constraint Logic Programming". In: *Constraints* 4 (Feb. 1999), pp. 43–65. DOI: `10.1023/A:1009845710839`.

[6] Toby Walsh. *026: Sports Tournament Scheduling.* URL: `https://www.csplib.org/Problems/prob026/`. (accessed: 01.09.2025).