Modelling and solving the Sports Tournament Schediling (STS) problem

Luwam Major Kefali (luwammajor.kefali@studio.unibo.it)
Hilina Fissha Woreta (hilina.woreta@studio.unibo.it)

1 Introduction

This report studies the Sports Tournament Scheduling (STS) problem using Constraint Programming (CP), propositional SAT, Satisfiability Modulo Theories (SMT), and Mixed-Integer Programming (MIP). This section formalizes the elements common to all models.

Input parameters. The input consists of an even integer n denoting the number of teams. The tournament is played over $W = n - 1$ weeks, each divided into $P = n/2$ periods, with exactly one match per period.

Round-robin preprocessing. All models employ a preprocessing step that fixes the match pairings in advance using the standard round-robin *circle method*. This classical construction generates a valid round-robin tournament for an even number of teams by fixing one team and cyclically rotating the remaining ones, ensuring that each team plays exactly once per week and that every pair of teams meets exactly once. The circle method is a well-established technique in sports scheduling and is described in the literature by de Werra [1] and Rasmussen [2]. Subsequent surveys confirm its continued use as a preprocessing step in sports scheduling models [3, 4].

By fixing the pairings beforehand, the solving phase is restricted to assigning matches to periods and, in the optimization variant, deciding the home and away roles.

Decision and optimization variants. After preprocessing, the decision problem requires assigning each match to a period such that no team appears more than twice in the same period over the whole tournament. An optional optimization variant is also considered, whose goal is to balance the number of home and away games per team by minimizing the maximum deviation from perfect balance.

Experimental setting. All approaches are evaluated under a uniform time limit of 300 seconds per instance, including preprocessing time.

2 CP Model

This section presents the Constraint Programming (CP) formulation of the Sports Tournament Scheduling (STS) problem. The model is implemented in *MiniZinc* and executed with a time limit of 300 seconds using the *Gecode*, *Chuffed*, and MiniZinc *CP-SAT* backends.

The round-robin pairings are generated in a preprocessing phase using a Python implementation of the circle method and passed as fixed input parameters to the MiniZinc model and the CP model decides only the assignment of matches to periods and, in the optimization variant, the home/away orientation.

2.1 Decision variables

For each week $w \in Weeks$ and each match $m \in Matches$, the model defines an integer decision variable

$$per[w, m] \in Periods,$$

where $per[w, m] = p$ means that match m of week w is scheduled in period p.

Optimization variables. In the fairness optimization variant, an additional Boolean variable is introduced:

$$flip[w, m] \in \{0, 1\},$$

where $flip[w, m] = 0$ keeps the ordering $(pair[w, m, 1], pair[w, m, 2])$, while $flip[w, m] = 1$ swaps the two teams, thus determining the home and away roles. An auxiliary variable $home[w, m] \in Teams$ is derived from $flip[w, m]$ and is used for counting home appearances via a global cardinality constraint.

2.2 Objective function

In the optimization variant, the objective is to balance home and away games for each team.

Home team variables. To express fairness counting with stronger propagation, we define an auxiliary variable for the home team of each match:

$$home[w, m] \in Teams,$$

linked to $flip[w, m]$ by:

$$home[w, m] = \begin{cases} pair[w, m, 1] & \text{if } flip[w, m] = 0, \\ pair[w, m, 2] & \text{if } flip[w, m] = 1. \end{cases}$$

Home-game counts via global cardinality. Let $H = \{home[w, m] \mid w \in Weeks,\, m \in Matches\}$ be the multiset of home-team values across all matches. For each team $t \in Teams$ we define:

$$home\_games[t] = |\{(w, m) : home[w, m] = t\}|.$$

This is enforced in MiniZinc using a global cardinality constraint over the flattened array $[home[w, m]]$.

Objective. Since $W = n - 1$ is odd for even n, perfect balance is impossible. We therefore minimize the maximum deviation from balance:

$$max\_dev = \max_{t \in Teams} \big|2 \cdot home\_games[t] - W\big|, \qquad \min\ max\_dev.$$

The objective variable is bounded by $1 \le max\_dev \le W$.

Output. The reported schedule uses $per[w, m]$ to place matches into periods and uses $flip[w, m]$ to orient each match so that the first entry is the home team.

2.3 Constraints

C1: One match per period per week. Each period is used exactly once in every week:

$$\forall w \in Weeks : \texttt{alldifferent}\big(\{per[w, m] \mid m \in Matches\}\big).$$

C2: At most twice per period per team. Each team may appear in the same period at most twice over the whole tournament:

$$\forall t \in Teams,\ \forall p \in Periods : \sum_{w \in Weeks} \sum_{m \in Matches} \mathbf{1}\big(per[w, m] = p \wedge (pair[w, m, 1] = t \vee pair[w, m, 2] = t)\big) \le 2.$$

Implicit constraints. The constraints that each team plays exactly once per week and that every pair of teams meets exactly once are guaranteed by the fixed round-robin pairings and are therefore not explicitly encoded in the CP model.

Symmetry breaking constraints. Period labels are interchangeable. To break this symmetry, an optional constraint fixes the assignment of matches to periods in the first week:

$$\forall m \in Matches: \ per[1, m] = m.$$

This constraint is enabled when the parameter `use_sb` is set to 1.

Search strategy When enabled (`use_ss = 1`), a custom search strategy is applied:

$$\texttt{int\_search}(\{per[w, m]\}, \texttt{first\_fail}, \texttt{indomain\_min}),$$

which selects the most constrained variable first and assigns the smallest available value. Otherwise, the default solver search is used.

2.4 Validation

2.4.1 Experimental design

The CP model was implemented in *MiniZinc* and evaluated using three solver backends: *Gecode*, *Chuffed*, and the MiniZinc *CP-SAT* solver (labeled as `cp`). Experiments were conducted on instances with an even number of teams $n \in \{6, 8, 10, 12, 14, 16, 18, 20, 22\}$. A uniform time limit of 300 seconds was imposed for each instance.

To assess the impact of modeling choices, we evaluated multiple configurations of the model by varying:

1. the problem variant (decision vs. fairness optimization),
2. the use of symmetry breaking constraints, and
3. the presence or absence of a user-defined search strategy.

Symmetry breaking fixes the assignment of periods in the first week, thereby removing permutations of period labels. When enabled, the custom search applies a `first_fail` variable ordering with `indomain_min` value selection over the period-assignment variables (and the home/away variables in the optimization variant).

Reproducibility and setup. All experiments were run on a Lenovo laptop equipped with an Intel Core i5 processor and 8 GB of RAM, using MiniZinc version 2.9.4. Each run was executed in a single-process setting with no external parallelism. No randomization is used in the model or search strategies; therefore, all experiments are fully deterministic given the same solver versions and execution parameters.

2.4.2 Experimental results

The experimental results are reported in Tables 1–4. For decision problems, table entries show runtime in seconds. For optimization problems, entries report the best objective value found, with boldface indicating proven optimality. The label "NA" denotes instances for which no solution was found within the time limit.

N	Chuffed	Chuffed+SB	CP-SAT	CP-SAT+SB	Gecode	Gecode+SB
6	0	0	0	0	0	0
8	0	0	0	0	0	0
10	0	0	0	0	0	0
12	0	0	0	0	0	0
14	0	0	0	0	45	0
16	64	2	60	2	NA	NA
18	NA	NA	NA	NA	NA	NA
20	NA	117	NA	105	NA	NA
22	NA	NA	NA	NA	NA	NA

Table 1: Decision variant: runtime in seconds without custom search.

N	Chuffed	Chuffed+SB	CP-SAT	CP-SAT+SB	Gecode	Gecode+SB
6	**1**	**1**	**1**	**1**	**1**	**1**
8	**1**	**1**	**1**	**1**	**1**	**1**
10	**1**	**1**	**1**	**1**	**1**	**1**
12	**1**	**1**	**1**	**1**	**1**	**1**
14	9	**1**	13	**1**	**1**	**1**
16	NA	15	NA	15	NA	NA
18	NA	NA	NA	NA	NA	NA
20	NA	NA	NA	NA	NA	NA
22	NA	NA	NA	NA	NA	NA

Table 2: Optimization variant: best objective value (bold = proven optimal).

N	Chuffed+SS	Chuffed+SB+SS	CP-SAT+SS	CP-SAT+SB+SS	Gecode+SS	Gecode+SB+SS
6	0	0	0	0	0	0
8	0	0	0	0	0	0
10	0	0	0	0	0	0
12	0	0	0	0	0	0
14	NA	NA	NA	NA	10	10
16	NA	NA	NA	NA	NA	NA
18	NA	NA	NA	NA	NA	NA
20	NA	NA	NA	NA	NA	NA
22	NA	NA	NA	NA	NA	NA

Table 3: Decision variant: runtime in seconds with custom search.

N	Chuffed+SS	Chuffed+SB+SS	CP-SAT+SS	CP-SAT+SB+SS	Gecode+SS	Gecode+SB+SS
6	**1**	**1**	**1**	**1**	**1**	**1**
8	**1**	**1**	**1**	**1**	**1**	**1**
10	**1**	**1**	**1**	**1**	**1**	**1**
12	**1**	**1**	**1**	**1**	**1**	**1**
14	NA	NA	NA	NA	**1**	**1**
16	NA	NA	NA	NA	NA	NA
18	NA	NA	NA	NA	NA	NA
20	NA	NA	NA	NA	NA	NA
22	NA	NA	NA	NA	NA	NA

Table 4: Optimization variant with custom search: best objective value.

Discussion CP is very effective on small and medium instances, and symmetry breaking is the main performance lever: it turns several timeouts into solves (e.g., $n = 16$ and $n = 20$), while without SB the model quickly stalls beyond $n = 16$. Custom search did not help here and often made performance worse, suggesting that the solver's default heuristics already fit the model structure well, or our hearistics was lackluster.

3 SAT Model

This section presents a *pure SAT* encoding of the Sports Tournament Scheduling (STS) problem. The model is expressed entirely in propositional logic, translated into CNF [5] and exported in DIMACS format. All instances are solved using the Glucose SAT solver within a time limit of 300 seconds(including the DIMACs generation). As specified in section 1, a preprocessing phase is used to fix the round-robin structure.The SAT encoding is therefore responsible only for assigning these fixed matches to periods, while enforcing the remaining STS constraints.

3.1 Decision variables

Let n be even. We define the index sets:

$$W = \{0, \ldots, n-2\} \quad \text{(weeks)}, \qquad P = \{0, \ldots, \tfrac{n}{2}-1\} \quad \text{(periods)}, \qquad M = \{0, \ldots, \tfrac{n}{2}-1\} \quad \text{(matches per week)}.$$

Using the circle method, for each week $w \in W$ a fixed list of matches

$$\mathsf{pair}(w, m) = (a, b), \qquad m \in M,$$

is generated, where teams a and b play against each other in week w. These pairings are treated as input data and are not part of the SAT decision process.

The SAT variables encode only the assignment of matches to periods. For every week $w \in W$, match $m \in M$, and period $p \in P$, we introduce a Boolean variable:

$$X_{w,m,p} = \begin{cases} \texttt{True} & \text{iff match } m \text{ of week } w \text{ is assigned to period } p, \\ \texttt{False} & \text{otherwise.} \end{cases}$$

No variables are introduced for home–away decisions. In the decoded schedule, the orientation of each match is taken directly from the fixed ordering (a, b) produced by the circle method.

3.2 Objective function

There is no objective function in this SAT formulation. Glucose is used purely as a feasibility solver for the decision version of the STS problem, i.e., the goal is to find an assignment to the variables $X_{w,m,p}$ satisfying all constraints.

3.3 Constraints

All constraints are encoded directly in CNF over the variables $X_{w,m,p}$.

C1: Each match is assigned to exactly one period. For every week $w \in W$ and match $m \in M$, exactly one period must be selected:

$$\sum_{p \in P} X_{w,m,p} = 1.$$

This constraint is encoded using one clause enforcing "at least one" and pairwise clauses enforcing "at most one".

C2: Each period contains exactly one match per week. For every week $w \in W$ and period $p \in P$, exactly one match is assigned to that period:

$$\sum_{m \in M} X_{w,m,p} = 1.$$

This ensures a bijection between matches and periods within each week.

C3: Period frequency constraint. Each team may appear in the same period at most twice over the entire tournament. For a fixed team t and period p, let $m_t(w)$ denote the unique match index in week w involving team t. We impose:

$$\sum_{w \in W} X_{w,m_t(w),p} \leq 2.$$

This cardinality constraint is encoded in CNF by forbidding any triple of distinct weeks in which team t would be scheduled in period p. For all $w_1 < w_2 < w_3$:

$$(\neg X_{w_1,m_t(w_1),p} \ \vee \ \neg X_{w_2,m_t(w_2),p} \ \vee \ \neg X_{w_3,m_t(w_3),p}).$$

Implicit constraints

The constraints that each team plays exactly once per week and that each pair of teams meets exactly once are guaranteed by the round-robin preprocessing and are therefore not explicitly encoded in SAT.

Symmetry breaking

The problem admits symmetries due to the interchangeability of periods. When symmetry breaking is enabled, a canonical assignment is imposed by fixing an *anchor week* $aw \in W$ and enforcing:

$$X_{aw,m,m} = \texttt{True}, \qquad \forall m \in M.$$

This removes all symmetries induced by permuting period labels without excluding any essentially different solution.

3.4 Validation

3.4.1 Experimental design

The SAT model is implemented by generating the DIMACS CNF encoding in a preprocessing step and solving it with Glucose. The reported runtime includes both CNF generation and SAT solving time. Satisfying assignments are decoded into the required $(\frac{n}{2}) \times (n-1)$ schedule format. All experiments were run on a Lenovo laptop equipped with an Intel Core i5 processor and 8 GB of RAM.

3.4.2 Experimental results

N	Glucose	Glucose + SB
6	0	0
8	0	0
10	0	0
12	0	0
14	0	0
16	5	0
18	54	3
20	NA	NA

Table 5: SAT decision variant: runtime in seconds (300s timeout).

Discussion The pure SAT encoding scales well up to moderate sizes, solving up to $n = 18$, but is highly sensitive to symmetry breaking: the anchor-week choice can shift runtimes by orders of magnitude (e.g., $n = 18$ drops from 54s to 3s). However, for $n = 20$ the encoding remains too hard for Glucose within 300s, indicating that further (safe) symmetry breaking or stronger propagation would be needed.

4 SMT Model

This section presents the SMT formulation used in our implementation. Two encodings are considered: (i) an in-memory Z3 [7] model based on Boolean variables with pseudo-Boolean (PB) constraints, and (ii) an SMT-LIB2 exporter for external solvers (*cvc5 [8]* and *OpenSMT*), using integer period variables. All runs use a 300-second time limit and include preprocessing and encoding time in the reported runtime.

As in the CP and SAT approaches, the round-robin pairings are fixed in a preprocessing step using the circle method. The SMT encodings therefore decide only the assignment of the fixed weekly matches to periods and, in the optional fairness variant, the home/away orientation.

4.1 Decision variables

Let n be even, and define:

$$W = \{0, \dots, n-2\}, \qquad P = \{0, \dots, \tfrac{n}{2} - 1\}, \qquad M = \{0, \dots, \tfrac{n}{2} - 1\}.$$

For each week $w \in W$ and match $m \in M$, the circle method yields a fixed pairing $\mathsf{pair}(w, m) = (a, b)$.

Z3 (Boolean + PB encoding). For each $w \in W$, $m \in M$, $p \in P$, we introduce a Boolean variable

$$X_{w,m,p} \in \{\texttt{False}, \texttt{True}\},$$

where $X_{w,m,p} = \texttt{True}$ iff match m of week w is assigned to period p. For the fairness variant, an additional Boolean variable is introduced:

$$home_{w,m} \in \{\texttt{False}, \texttt{True}\},$$

where $home_{w,m} = \texttt{True}$ selects the first team of $\mathsf{pair}(w, m)$ as the home team; otherwise the second team is home.

External solvers (SMT-LIB2 with integers). For each $w \in W$ and $m \in M$, we introduce an integer variable

$$per_{w,m} \in \{0, \dots, |P| - 1\},$$

denoting the period of match (w, m). For fairness runs we additionally define a Boolean $home_{w,m}$ with the same meaning as above.

4.2 Objective function

The fairness objective aims to balance home and away games per team. Let $hg(t)$ be the number of home games of team t. Using the fixed pairings and the variables $home_{w,m}$, $hg(t)$ is defined as a linear sum of indicator terms over weeks.

We introduce an integer variable D and minimize it:

$$\min D$$

subject to the per-team imbalance constraints:

$$\big|2 \cdot hg(t) - |W|\big| \le D \qquad \forall t.$$

Since $|W| = n - 1$ is odd (for even n), perfect balance is impossible, hence $D \ge 1$. Also $hg(t) \in [0, |W|]$ implies $|2hg(t) - |W|| \le |W|$, therefore:

$$1 \le D \le |W|.$$

In the Z3 implementation, this objective is handled using Z3's optimization interface (`Optimize`).

For external solvers (cvc5, OpenSMT), the SMT-LIB2 exporter supports the same fairness constraints for a fixed bound D, and the smallest satisfiable bound is obtained by incrementally checking feasibility for $D = 1, 2, \dots$ up to a user-defined limit.

4.3 Constraints

All constraints are derived from the fixed round-robin pairings.

C1: Exactly one period per match. For every week $w \in W$ and match $m \in M$:

$$\sum_{p \in P} X_{w,m,p} = 1$$

(Z3 PB form), or equivalently for external solvers:

$$0 \leq per_{w,m} < |P|.$$

C2: Exactly one match per period per week. For every week $w \in W$ and period $p \in P$:

$$\sum_{m \in M} X_{w,m,p} = 1$$

or, in the integer encoding, all matches within a week must occupy distinct periods:

$$\mathsf{distinct}(per_{w,0}, \ldots, per_{w,|P|-1}).$$

C3: Team-period frequency constraint (with implied structure). Let $m_t(w)$ denote the unique match index in week w that involves team t. For each team t and period p, define the number of occurrences:

$$occ(t,p) = \sum_{w \in W} \mathbf{1}\big(X_{w,m_t(w),p}\big) \quad \text{(Boolean encoding)}$$

or

$$occ(t,p) = \sum_{w \in W} \mathbf{1}\big(per_{w,m_t(w)} = p\big) \quad \text{(integer encoding)}.$$

The base STS constraint requires $occ(t,p) \leq 2$.

In our SMT implementations, we additionally enforce the implied structure:

$$1 \leq occ(t,p) \leq 2 \qquad \forall t, \forall p,$$

and

$$\sum_{p \in P} \mathbf{1}\big(occ(t,p) = 1\big) = 1 \qquad \forall t.$$

Intuitively, since $\sum_p occ(t,p) = |W| = n-1$ and each $occ(t,p) \leq 2$, each team must have exactly one period where it appears once, and it appears twice in all other periods. This implied structure is encoded explicitly in both the Z3 PB model and the SMT-LIB2 exporter.

Fairness constraint (bounded feasibility). For a given bound D, for each team t we define the number of home games $hg(t)$ using the Boolean variables $home_{w,m}$ associated to the match containing t in week w. The model enforces:

$$|2 \cdot hg(t) - |W|| \leq D \qquad \forall t.$$

In SMT-LIB2 this is encoded using two linear inequalities:

$$2 \cdot hg(t) - |W| \leq D, \qquad |W| - 2 \cdot hg(t) \leq D.$$

Symmetry breaking

Period symmetry. Period labels are interchangeable. When enabled, symmetry breaking fixes week 0 diagonally:

$$X_{0,m,m} = \texttt{True} \quad \forall m \in M,$$

equivalently $per_{0,m} = m$ in the integer encoding.

Pinned matches. An additional optional constraint pins the match involving team 1 to period 0 for the first k weeks:

$$X_{w,m_1(w),0} = \texttt{True} \quad \forall w \in \{0, \ldots, k-1\},$$

equivalently $per_{w,m_1(w)} = 0$.

Home/away symmetry (fairness runs only). When fairness variables are enabled, flipping all $home_{w,m}$ values yields an equivalent solution. This symmetry is broken by fixing one orientation:

$$home_{0,0} = \texttt{True}.$$

4.4 Validation

4.4.1 Experimental design

All SMT experiments were run with a 300-second time limit. For the Z3 encoding, the solver is invoked through the Z3 Python API and configured with a 300 s timeout. For external solvers, an SMT-LIB2 file is generated and solved using *cvc5* and *OpenSMT*. Reported runtimes include preprocessing and encoding time. All experiments were run on a Lenovo laptop equipped with an Intel Core i5 processor and 8 GB of RAM.

Experimental results The following tables report the observed runtimes (seconds). "NA" indicates that no satisfying model was obtained within the time limit or that the run was not completed for that configuration.

N	Z3	Z3 + Pin(k=1)	Z3 + SB	Z3 + SB + Pin(k=1)
6	0	0	0	0
8	0	0	0	0
10	0	0	0	0
12	0	0	0	0
14	0	0	0	0
16	1	1	3	0
18	110	22	21	24
20	267	81	215	28
22	148	NA	NA	NA

Table 6: Z3 decision variant: runtime in seconds (300s timeout).

N	Z3 Opt ($D^\star$)	Z3 Opt + Pin(k=1) ($D^\star$)	Z3 Opt + SB ($D^\star$)	Z3 Opt + SB + Pin(k=1) ($D^\star$)
6	**1**	**1**	**1**	**1**
8	**1**	**1**	**1**	**1**
10	**1**	**1**	**1**	**1**
12	**1**	**1**	**1**	**1**
14	**1**	**1**	**1**	**1**
16	**1**	**1**	**1**	**1**
18	**1**	**1**	**1**	**1**
20	**1**	**1**	**1**	**1**
22	NA	NA	NA	**1**

Table 7: Z3 optimization variant: smallest value found $D^\star$ (300s timeout).

N	cvc5	cvc5 + Pin(k=1)	cvc5 + SB	cvc5 + SB + Pin(k=1)
6	0	0	0	0
8	2	1	2	2
10	48	6	18	20
12	120	128	146	148
14	NA	NA	NA	NA
16	NA	NA	NA	NA
18	NA	NA	NA	NA
20	NA	NA	NA	NA
22	NA	NA	NA	NA

Table 8: External SMT solvers decision variant (CVC5): runtime in seconds (300s timeout).

N	OpenSMT	OpenSMT + Pin(k=1)	OpenSMT + SB	OpenSMT + SB + Pin(k=1)
6	0	0	0	0
8	0	0	0	0
10	10	8	13	11
12	20	90	76	78
14	NA	NA	NA	NA
16	NA	NA	NA	NA
18	NA	NA	NA	NA
20	NA	NA	NA	NA
22	NA	NA	NA	NA

Table 9: External SMT solvers decision variant (OpenSMT): runtime in seconds (300s timeout).

Discussion SMT with Z3 is the most scalable approach in our experiments: with lightweight symmetry breaking and/or pinning it solves feasibility up to $n = 22$, and the added implied structure constraints noticeably improve robustness for larger n. In contrast, the external SMT solvers (cvc5 and OpenSMT) solve only up to $n = 12$ in our integer encoding, showing that solver–encoding interactions dominate performance at larger sizes.

5 MIP Model

This section presents the Mixed-Integer Linear Programming (MIP) formulation used for the Sports Tournament Scheduling (STS) problem. All models share the global definitions introduced in Section 1, including the sets of teams, weeks and periods, as well as the round-robin preprocessing step.

Four variants of the MIP formulation are considered:

- a plain feasibility model,
- a symmetry-breaking model,
- a model with implied constraints,
- an optimization model minimizing home/away imbalance.

All models are linear and solver-independent.

5.1 Decision variables

All MIP variants use the binary decision variable

$$x_{ijwp} = 1 \iff \text{the match between teams } i \text{ and } j \text{ in week } w \text{ is assigned to period } p,$$

for all preprocessed matches $\{i, j\}$, weeks w, and periods p.

To model home and away assignments, the optimization model introduces the additional binary variable

$$y_{ijwp} = 1 \iff \text{team } i \text{ plays at home and team } j \text{ plays away in week } w, \text{ period } p.$$

The following auxiliary integer variables are used in the optimization model:

$$h_t,\, a_t,\, d_t \in \mathbb{Z}_{\geq 0} \quad \forall t, \qquad F \in \mathbb{Z}_{\geq 0},$$

where h_t and a_t represent the number of home and away matches of team t, d_t its home/away imbalance, and F the maximum imbalance among all teams.

5.2 Objective function

Decision models. The plain, symmetry-breaking and implied-constraint variants address the feasibility (decision) version of the STS problem and therefore use a dummy objective function:

$$\min 0.$$

Fairness optimization model. The optimization variant aims at balancing home and away matches by minimizing the maximum imbalance over all teams:

$$\min F.$$

The imbalance is linearised through the constraints

$$d_t \geq h_t - a_t, \qquad d_t \geq a_t - h_t, \qquad F \geq d_t \quad \forall t.$$

5.3 Constraints

All MIP variants include the following linear constraints.

Exactly one match per period and week. Each period of every week hosts exactly one match:

$$\sum_{\{i,j\} \in M_w} x_{ijwp} = 1 \quad \forall w, p,$$

where M_w denotes the set of matches scheduled in week w by the round-robin preprocessing.

Weekly participation constraint. Each team plays exactly one match per week:

$$\sum_{\substack{\{i,j\} \in M_w \\ i=t \vee j=t}} \sum_{p} x_{ijwp} = 1 \quad \forall t, w.$$

Period frequency constraint. Each team appears in the same period at most twice over the tournament:

$$\sum_{w} \sum_{\substack{\{i,j\} \in M_w \\ i=t \vee j=t}} x_{ijwp} \leq 2 \quad \forall t, p.$$

Orientation consistency. A home/away orientation can only be assigned if the corresponding match is scheduled:

$$y_{ijwp} \leq x_{ijwp} \quad \forall i, j, w, p.$$

Home and away counting. The total number of home and away matches of each team is computed by aggregating the orientation variables across all weeks and periods. Since every scheduled match contributes exactly one home team and one away team, the total number of home matches equals the total number of away matches implicitly.

Symmetry-breaking constraints

To reduce symmetries due to interchangeable period labels, a symmetry-breaking constraint is imposed by fixing the match involving team 1 in week 1 to be assigned to period 1:

$$\sum_{\substack{\{i,j\} \in M_1 \\ i=1 \vee j=1}} x_{ij11} = 1,$$

where M_1 denotes the set of matches scheduled in week 1. This constraint eliminates equivalent solutions obtained by permuting periods without removing any feasible schedules.

Implied constraints

To strengthen the formulation, an implied constraint is added stating that no team can appear in the same period in three consecutive weeks:

$$\sum_{\substack{\{i,j\} \in M_{w,w+1,w+2} \\ i=t \vee j=t}} x_{ijw'p} \leq 2 \quad \forall t, \ \forall p, \ \forall w = 1, \ldots, n-3,$$

where $M_{w,w+1,w+2}$ denotes the set of matches scheduled in weeks $w, w+1$, and $w+2$, and $w' \in \{w, w+1, w+2\}$.

This constraint is redundant with respect to feasibility, as each team can appear in a given period at most twice over the entire tournament, but it tightens the linear relaxation and improves solver performance.

5.4 Validation

5.4.1 Experimental design

All MIP models were implemented in AMPL and evaluated using the MIP solvers CPLEX and Gurobi with a time limit of 300 seconds and all experiments were run on a 64-bit machine with an Intel Core Ultra 7 256V CPU (2.20 GHz) and 16 GB of RAM. We tested instances with $n \in \{6, 8, 10, 12, 14, 16, 18, 20, 22, 24\}$. For each instance, we recorded solver runtime for the decision models and the best objective value for the optimization model. When no solution was returned within the time limit, the entry is marked as `NA`. **The time required to generate the round-robin structure and load the instance into AMPL was measured and found to be negligible (below 0.01 seconds for all tested instances), and therefore has no impact on the reported runtimes.**

5.4.2 Experimental results

Decision model results. Table 10 reports the solving times for the three feasibility models (plain, symmetry-breaking, and implied-constraints), where times are rounded down to the nearest second and `NA` denotes a timeout.

Table 10: Decision models (time in seconds, 300 → NA)

N	MIP_implied_cplex	MIP_implied_gurobi	MIP_plain_cplex	MIP_plain_gurobi	MIP_symmetry_cplex	MIP_symmetry_gurobi
6	0	0	0	0	0	0
8	0	0	0	0	0	0
10	0	0	0	0	0	0
12	0	0	0	1	0	0
14	0	9	0	4	0	1
16	112	17	9	NA	106	NA
18	42	NA	NA	NA	41	NA
20	NA	NA	27	NA	NA	NA
22	NA	NA	NA	NA	NA	266
24	NA	NA	NA	NA	NA	NA

Optimization model results. Table 11 reports the optimal value of the fairness objective F. Entries marked as `NA` indicate that no feasible solution was found within the time limit.

Table 11: Optimization model — optimal fairness values

N	MIP_opt_cplex	MIP_opt_gurobi
6	1.0	1.0
8	1.0	1.0
10	1.0	1.0
12	1.0	1.0
14	1.0	NA
16	1.0	NA
18	NA	NA
20	NA	NA
22	NA	NA
24	NA	NA

5.5 Discussion

The results show that the difficulty of the STS decision problem increases rapidly with the number of teams and that the effectiveness of modelling choices depends on both the instance size and the solver.

All formulations are solved almost instantaneously for small instances ($n \leq 12$). From $n = 14$ onward, performance differences become evident. The *plain* model can still solve some medium-sized instances (e.g. $n = 20$ with CPLEX), but its behaviour is inconsistent. The *symmetry-breaking* model improves robustness and is the only variant that solves the instance with $n = 22$, albeit close to the time limit. The *implied-constraints* model strengthens the formulation but does not consistently outperform the other variants for larger instances.

Overall, no single decision formulation dominates, and solver–model interactions play a significant role.

Optimization model. The fairness objective substantially increases problem difficulty. For all solved instances, the optimal value is $F = 1.0$, but feasible solutions are only found for small instances, with CPLEX solving up to $n = 16$ and Gurobi up to $n = 12$. For larger instances, neither solver returns a solution within the time limit, highlighting the limited scalability of the optimization formulation.

6 Conclusions

We compared CP, SAT, SMT, and MIP formulations for the STS problem under a uniform preprocessing scheme (fixed round-robin pairings) and a 300-second time limit.

Overall, SMT (Z3) is the most scalable feasibility approach in our experiments, solving up to $n = 22$ with simple symmetry breaking and pinning. CP is strong on small and medium instances and, with symmetry breaking, reaches $n = 20$, but does not remain robust beyond that in our tested configurations. The pure SAT encoding is competitive up to $n = 18$, yet performance is extremely sensitive to symmetry breaking and $n = 20$ remains unsolved within the time limit. MIP provides a clean, solver-independent formulation and performs well on smaller instances, but becomes inconsistent on larger n, and the fairness optimization variant is substantially harder across all paradigms.

Authenticity and Author Contribution Statement

The work is our own and we did not copy it from someone else. Ideas that we took from literature search and existing implementations are cited properly. We made use of AI tools for report writing and formatting, and sporadically to debug coding errors.

Author's contribution to the work:
The modelling choices were decided together. Then division was carried out for the implementationas follows: Luwam/Hilina jointly handled CP, Luwam handled SAT and SMT, while Hilina handled MIP approach.

References

[1] D. de Werra. Scheduling in sports. In *Studies on Graphs and Discrete Programming*, pages 381–395, 1981.

[2] R. Rasmussen. *Scheduling Tournaments*. Springer, 2008.

[3] G. Kendall, S. Knust, C. C. Ribeiro, and S. Urrutia. Scheduling in sports: An annotated bibliography. *Computers & Operations Research*, 37(1):1–19, 2010.

[4] M. A. Trick. Sports scheduling. In *Handbook of Combinatorial Optimization*. Springer, 2012.

[5] C. Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *CP 2005*.

[6] J.-F. Puget. Symmetry Breaking Revisited. In *CP 2005*.

[7] L. M. de Moura and N. Bjørner. Z3: An Efficient SMT Solver. In *TACAS 2008*.

[8] A. Reynolds et al. cvc5: A Next-Generation SMT Solver. In *CAV 2020*.