Modelling and Solving the Sports Tournament Scheduling (STS) Problem

Alessia Bedeschi - alessia.bedeschi@studio.unibo.it
Julian Pajo - julian.pajo@studio.unibo.it

September 2025

1 Introduction

This report addresses the Sports Tournament Scheduling (STS) problem for n teams, played over $n-1$ weeks with $n/2$ periods per week. In each period, two teams play a game, one at home and one away.

1.1 Input Parameters

The model takes as input a single parameter n, representing the total number of teams. From this value we derive the set of teams $T = \{1, \ldots, n\}$, the set of weeks $W = \{1, \ldots, n-1\}$, and the set of periods per week $P = \{1, \ldots, n/2\}$.

1.2 Decision Variables

The decision variables represent the matches between two teams, one playing at home and the other away, across all weeks and periods. Their specific encoding differs from one solver to another.

1.3 Optimization Objective

The goal is to balance the number of games at home (H_i) and away (A_i) for each team, ensuring fairness in the tournament. With an even number of teams, each plays $n-1$ matches. Therefore, a perfect home/away balance is impossible: the minimum imbalance is **1**, e.g. $(\frac{n-1}{2} + 1)$ home vs. $(\frac{n-1}{2})$ away.
The maximum occurs if always home or always away: $H_i = n-1$, $A_i = 0$, hence $|H_i - A_i| = n - 1$.
Therefore, the imbalance for a team satisfies the following.

$$1 \ \leq \ |H_i - A_i| \ \leq \ n - 1.$$

1.4 Constraints

The problem constraints common to all models are:

1. Each team plays exactly once per week.
2. Each team plays with every other team exactly once.
3. Each team plays at most twice in the same period over the tournament.

1.4.1 Implied Constraints

Several implied constraints can be exploited to reduce the search space and improve solver efficiency.

- **No self-matches:** Each team must face every other exactly once, so it cannot face itself. Explicitly forbidding this reduces the search space.
- **Symmetry of matches:** If team i plays j in a week, then j must play i, with exactly one at home. This reinforces uniqueness constraints and detects inconsistencies earlier.
- **Period consistency:** opponents in the same week must be assigned to the same period, ensuring coherence between match and period variables.
- **Two teams per period:** Each period corresponds to exactly one match, which requires exactly two teams.

1.4.2 Symmetry Breaking Constraints

To reduce the search space and improve solver efficiency, we exploit symmetry-breaking constraints inspired by the study of Regin [1]. According to Regin, two types of symmetries are present in the tournament scheduling problem:

1. **Home-away symmetry:** All home games can be replaced by away games and vice versa.
2. **Team permutation symmetry:** Teams can be permuted without affecting the validity of the schedule.

The first type of symmetry is removed by enforcing that the first game of the first team is a home game. The second type is addressed by fixing the schedule of the first team, for instance, by assigning opponents in increasing order.

2 CP Model

2.1 Decision variables

In the CP model, each match is represented by the following decision variables:

- $O_{i,j} \in \{1, \dots, n\} \setminus \{i\}$: the opponent of team i in week j.
- $PL_{i,j} \in \{0, 1\}$: the venue of the match for team i in week j, where 0 represents an away game and 1 a home game.
- $\mathrm{per}(i, j) \in \{1, \dots, n/2\}$: the period or slot in which team i plays in week j.

The formalization follows the framework of Régin [1].

2.2 Objective function

The goal is to minimize the imbalance between home and away games. For each team $i \in \{1, \dots, n\}$ we compute the numbers of home and away games:

$$H_i = \sum_{w \in W} (PL_{i,w} = 1), \qquad A_i = \sum_{w \in W} (PL_{i,w} = 0).$$

The imbalance of team i is:

$$\delta_i = |H_i - A_i|.$$

The optimization problem is therefore minimizing the global imbalance:

$$\min\ (\ \max_{i=1,\dots,n} \delta_i\ ).$$

We applied the bounds defined in Section 1.3.

2.3 Constraints

The constraints are divided into **base (hard) constraints**, **implied constraints**, and **symmetry-breaking constraints**. All formalizations refer back to the Introduction (Section 1.4).

2.3.1 Base constraints

- **C1** - Each team plays every other team exactly once:

$$\forall i \in T : \ \mathrm{allDifferent}\big(\{O_{i,w} \mid w \in W\}\big)$$

- **C2** - Each team plays exactly once per week:

$$\forall w \in W : \ \mathrm{allDifferent}\big(\{O_{i,w} \mid i \in T\}\big)$$

- **C3** - Teams occupy at most two matches in the same period (per):

$$\forall i \in T : \ \mathrm{gcc}\Big(\{\mathrm{per}(i, w) \mid w \in W\},\ P,\ L,\ U\Big)$$

  where:

  - $L = [0, 0, \dots, 0]$ is an array of length P with the lower bounds.
  - $U = [2, 2, \dots, 2]$ is an array of length P with the upper bounds.

2.3.2 Implied constraints

We define as follows the implied constraints described in Section 1.4.1:

- **No self-matches**:
$$\forall i, j : O_{i,j} \neq i$$
- **Symmetry of matches**:
$$\forall i, j : O_{O_{i,j},j} = i \land PL_{i,j} + PL_{O_{i,j},j} = 1$$
- **Period consistency:**
$$\forall i, j : \mathrm{per}(O_{i,j}, j) = \mathrm{per}(i, j)$$
- **Two teams per period per week**:
$$\forall w \in W : \ \mathrm{gcc}\Big(\{\mathrm{per}(t, w) \mid t \in T\},\ P,\ L,\ U\Big)$$
where:
  - $L = [2, 2, \ldots, 2]$ is an array of length P with the lower bounds.
  - $U = [2, 2, \ldots, 2]$ is an array of length P with the upper bounds.

2.3.3 Symmetry-breaking constraints

We exploit two main symmetries described in Section 1.4.2:

1. **Home-away symmetry**: We fix the first match of the first team as home:
$$PL_{1,1} = 1$$
2. **Team permutation symmetry**: We fix the schedule of the first team in increasing order of opponents:
$$\forall j \in W : O_{1,j} = j + 1$$

2.4 Validation

2.4.1 Experimental Design

All experiments were conducted inside a Docker container on an *Acer Nitro 5 AN515-58* equipped with an *Intel i7-12700H CPU* (2.70 GHz) and *16 GB RAM.* Two solvers were employed, *Gecode* and *Chuffed*, and four search strategies were evaluated: the default plain `solve`, the combination of `dom/wdeg` with `indomain_min`, the same with `restart_luby(250)`, and finally with `relax_and_reconstruct` (fixing 85% of the variables). Each run was restricted to 300 seconds. Both the decision (satisfiability) and optimization variants of the model were evaluated on instances with 6 to 16 teams (step size 2). For each strategy, solvers were tested with and without symmetry-breaking constraints to measure their effect on efficiency.

2.4.2 Experimental Results

The results are reported in the following tables, with one table for each search strategy and solver configuration. In the decision version, runtimes are expressed in seconds, while in the optimization version the best objective value found is reported, with optimal values highlighted in bold. Instances classified as `UNSAT` are unsatisfiable, whereas `N/A` indicates that no solution was found within the time limit.

ID	Gecode w/out SB	Gecode + SB	Chuffed w/out SB	Chuffed + SB
6	0	0	0	0
6 (opt)	**1**	**1**	**1**	**1**
8	0	0	0	0
8 (opt)	**1**	**1**	**1**	**1**
10	0	0	0	0
10 (opt)	**1**	**1**	**1**	**1**
12	N/A	0	8	0
12 (opt)	N/A	**1**	9	**1**
14	N/A	17	N/A	162
14 (opt)	N/A	11	N/A	N/A
16	N/A	N/A	N/A	N/A
16 (opt)	N/A	N/A	N/A	N/A

Table 1: Results using default search strategy.

ID	Gecode w/out SB	Gecode + SB	Chuffed w/out SB	Chuffed + SB
6	0	0	0	0
6 (opt)	**1**	**1**	**1**	**1**
8	0	0	0	0
8 (opt)	**1**	**1**	**1**	**1**
10	0	0	0	0
10 (opt)	**1**	**1**	**1**	**1**
12	0	1	0	0
12 (opt)	**1**	**1**	**1**	**1**
14	5	18	6	6
14 (opt)	**1**	**1**	**1**	**1**
16	220	N/A	6	5
16 (opt)	15	N/A	**1**	3

Table 2: Results using search strategy dom/wdeg + indomain_min.

ID	Gecode w/out SB	Gecode + SB	Chuffed w/out SB	Chuffed + SB
6	0	0	0	0
6 (opt)	**1**	**1**	**1**	**1**
8	0	0	0	0
8 (opt)	**1**	**1**	**1**	**1**
10	0	0	0	0
10 (opt)	**1**	**1**	**1**	**1**
12	0	0	0	0
12 (opt)	**1**	**1**	**1**	**1**
14	21	10	16	6
14 (opt)	**1**	**1**	**1**	**1**
16	N/A	N/A	N/A	103
16 (opt)	N/A	N/A	N/A	11

Table 3: Results using search strategy dom/wdeg + indomain_min + luby restart.

ID	Gecode w/out SB	Gecode + SB	Chuffed w/out SB	Chuffed + SB
6	0	0	0	0
6 (opt)	**1**	**1**	**1**	**1**
8	0	0	0	0
8 (opt)	**1**	**1**	**1**	**1**
10	0	0	0	0
10 (opt)	3	3	**1**	**1**
12	0	0	8	0
12 (opt)	5	3	9	**1**
14	21	11	N/A	162
14 (opt)	7	7	N/A	N/A
16	N/A	N/A	N/A	N/A
16 (opt)	N/A	N/A	N/A	N/A

Table 4: Results using search strategy dom/wdeg + indomain_min + luby restart + LNS.

The most effective strategy is **dom/wdeg + indomain_min**, which scales up to 16 teams, while the **default strategy** performs worst. **Symmetry breaking** generally improves runtimes, especially with Gecode. Conversely, **Luby restarts** and **LNS** introduce overhead and hinder performance. Overall, **Chuffed** consistently outperforms Gecode on larger instances.

3 SAT Model

3.1 Decision variables

The SAT model represents each match using the following Boolean decision variables:

- $home_{i,j,w} \in \{True, False\}$: $True$ if team i plays at home against team j in week w, $False$ otherwise.
- $per_{i,w,p} \in \{True, False\}$: $True$ if team i plays in period p during week w, $False$ otherwise.

3.2 Objective function

Optimization is performed using a binary search strategy, where each iteration introduces a constraint limiting the maximum allowable imbalance between home and away games, progressively guiding the solution towards a more balanced schedule.

3.3 Constraints

There are different categories of constraints used in the SAT model: **base constraints**, **channeling constraint** (to ensure consistency between match assignments and period allocations,**implied constraints**, **symmetry-breaking constraints** and **optimization constraint**.

3.3.1 Base constraints

- **C1** - Each team plays every other team exactly once:

$$\forall i, j \in T, i < j : \sum_{w \in W} (home_{i,j,w} + home_{j,i,w}) = 1$$

- **C2** - Each team plays exactly once per week:

$$\forall i \in T, \forall w \in W : \sum_{\substack{j \in T \\ j \neq i}} home_{i,j,w} + \sum_{\substack{j \in T \\ j \neq i}} home_{j,i,w} = 1$$

- **C3** - Teams occupy at most two matches in the same period (per):

$$\forall i \in T, \forall p \in P : \sum_{w \in W} per_{i,w,p} \leq 2$$

3.3.2 Channeling constraint

We add this constraint to maintain consistency and enable bidirectional propagation between the two formulations.

- **Match-period consistency**:

$$\forall i, j \in T, i < j, \forall w \in W: \left(home_{i,j,w} \vee home_{j,i,w}\right) \implies \sum_{p \in P}(per_{i,w,p} \wedge per_{j,w,p})$$

3.3.3 Implied constraints

The following are the implied constraints used in the model:

- **Exactly two teams must be assigned to each period in each week**:

$$\forall w \in W, \forall p \in P: \sum_{i \in T} per_{i,w,p} = 2$$

- **Mutual exclusion between home and away assignments**:

$$\forall i, j \in T, i \leq j, \forall w \in W: \neg home_{i,j,w} \vee \neg home_{j,i,w}$$

- **Each team must be assigned to exactly one period per week**:

$$\forall i \in T, \forall w \in W: \sum_{p \in P} per_{i,w,p} = 1$$

3.3.4 Optimization constraint

We define the optimization constraint as follows.

$$\forall i \in T: \text{min\_home}_i \leq \sum_{\substack{j \in \text{Teams} \\ j \neq i}} \sum_{w \in \text{Weeks}} home_{i,j,w} \leq \text{max\_home}_i$$

with

$$\text{min\_home}_i = \left\lfloor \frac{total\_games - \delta}{2} \right\rfloor, \quad \text{max\_home}_i = \left\lfloor \frac{total\_games + \delta}{2} \right\rfloor$$

considering $total\_games = len(T) - 1$ as the total number of games each team plays and δ as the maximum allowed imbalance

3.3.5 Symmetry-breaking constraints

We exploit two key symmetries, as discussed in Section 1.4.2 and add a symmetry-breaking constraint to impose only when we are not looking for optimization.

1. **Home-away symmetry**: We fix the first match of the first team as home:

$$home_{0,1,0} = True, \quad per_{0,0,0} = True, \quad per_{1,0,0} = True$$

2. **Team permutation symmetry**: We fix the schedule of the first team in increasing order of opponents:

$$\forall w \in W : \quad home_{0,w+1,w} + home_{w+1,0,w} = 1$$

3. **Home team must have smaller index than away team** (applied only when we are not optimizing the problem):

$$\forall i, j \in T, i > j, \forall w \in W : \neg home_{i,j,w}$$

3.4 Validation

3.4.1 Experimental Design

All experiments were run inside a Docker container on a HP Laptop 14s-dq4xxx with an Intel Core i5-1155G7 CPU (2.50 GHz) and 8GB RAM. The implementation was carried out using the Z3 solver. The model was translated into CNF in DIMACS format, allowing its resolution also with the Glucose SAT solver.

3.4.2 Experimental Results

The experimental results are summarized in the following table (as in Section 2.4.2):

ID	Z3 w/out SB	Z3 + SB	glucose w/out SB	glucose + SB
6	0	0	3	3
6 (opt)	**1**	**1**	**1**	**1**
8	0	0	9	8
8 (opt)	**1**	**1**	**1**	**1**
10	0	0	21	23
10 (opt)	**1**	**1**	**1**	**1**
12	2	1	39	38
12 (opt)	**1**	**1**	**1**	**1**
14	58	5	75	72
14 (opt)	**1**	**1**	7	7
16	N/A	9	123	290
16 (opt)	N/A	4	N/A	N/A
18	N/A	158	N/A	N/A
18 (opt)	N/A	N/A	N/A	N/A

Table 5: Results for SAT instances with and without symmetry breaking (SB).

Symmetry breaking constraints are crucial for Z3, allowing it to find solutions for larger instances (like 18 teams) that are unsolvable without them. However, for glucose these constraints show no significant effect.

4 SMT Model

4.1 Decision variables

In the SMT model, each match is represented by the following decision variables, created similarly to the one in the SAT model.

- $home_{i,j,w} \in \{True, False\}$: $True$ if team i plays at home against team j in week w, $False$ otherwise.
- $per_{i,w} \in \{1, ..., n/2\}$: team i plays during week w in period p.

4.2 Objective function

Again, the objective function to minimize is the maximum imbalance between home and away games for each team. As in SAT, we proceed with a binary search.

4.3 Constraints

The constraints used in the SMT model are the same as those used in the SAT model, modified according to the formulation of the decision variables.

4.3.1 Base constraints

- **C1** - Each team plays every other team exactly once:

$$\forall i, j \in T, i < j : \sum_{w \in W} (home_{i,j,w} + home_{j,i,w}) = 1$$

- **C2** - Each team plays exactly once per week:

$$\forall i \in T, \forall w \in W : \sum_{\substack{j \in T \\ j \neq i}} home_{i,j,w} + \sum_{\substack{j \in T \\ j \neq i}} home_{j,i,w} = 1$$

- **C3** - Teams occupy at most two matches in the same period (per):

$$\forall i \in T, \forall p \in P : \Big(\sum_{w \in W} per_{i,w} = p \Big) \leq 2$$

4.3.2 Channeling constraint

Again, we add this constraint to maintain consistency between $home_{i,j,w}$ and $per_{i,w}$

- **Match-period consistency**:

$$\forall i, j \in T, i < j, \forall w \in W: \ \big(home_{i,j,w} \vee home_{j,i,w}\big) \implies (per_{i,w} = per_{j,w})$$

4.3.3 Implied constraints

We define the implied constraints as follows:

- **Exactly two teams must be assigned to each period in each week**:

$$\forall w \in W, \forall p \in P: \ \Big(\sum_{i \in T} per_{i,w} = p\Big) = 2$$

- **Mutual exclusion between home and away assignments**:

$$\forall i, j \in T, i \leq j, \forall w \in W: \neg home_{i,j,w} \vee \neg home_{j,i,w}$$

4.3.4 Optimization constraint

The optimization constraint is like the one in the SAT model.

4.3.5 Symmetry-breaking constraints

We exploit two main symmetries described in Section 1.4.2 and add a symmetry-breaking constraint to impose only when we are not looking for optimization.

1. **Home-away symmetry**: We fix the first match of the first team as home:
$$home_{0,1,0} = True, \quad per_{0,0} = 0, \quad per_{1,0} = 0$$
2. **Team permutation symmetry**: We fix the schedule of the first team in increasing order of opponents:
$$\forall w \in W: \quad home_{0,w+1,w} + home_{w+1,0,w} = 1$$
3. **Home team must have smaller index than away team** (applied only when we are not optimizing the problem):
$$\forall i, j \in T, i > j, \forall w \in W: \neg home_{i,j,w}$$

4.4 Validation

4.4.1 Experimental Design

We adopt the same experimental setup as in the SAT model (Section 3.4.1). The formulation was implemented using Z3.

4.4.2 Experimental Results

The experimental results are summarized in the following table:

ID	Z3 w/out SB	Z3 + SB
6	0	0
6 (opt)	**1**	**1**
8	0	0
8 (opt)	**1**	**1**
10	1	0
10 (opt)	**1**	**1**
12	3	0
12 (opt)	3	**1**
14	N/A	2
14 (opt)	N/A	N/A
16	N/A	200
16 (opt)	N/A	N/A

Table 6: Results for SMT instances with and without symmetry breaking (SB).

Similarly to the SAT model, the results confirm that symmetry breaking constraints significantly improve the solver's performance for this problem when using the Z3 SMT solver. We adopted a different encoding for periods, using a single integer variable per team and week to represent the period assignment. Even if this integer formulation is more compact and straightforward to write, it proved to be less efficient for the solver. This is likely because reasoning about integers introduces more complex constraints and computational overhead for the solver compared to the purely Boolean reasoning used in the SAT encoding.

5 MIP Model

5.1 Decision variables

We introduce the following binary decision variables, defined for teams $i, k \in T$, weeks $w \in W$, and periods $p \in P$:

- **Match scheduling (unordered).** For each unordered pair $\{i, k\}$ with $i < k$ and each week w:

  $y_{i,k,w} \in \{0, 1\}, \quad y_{i,k,w} = 1 \;\Leftrightarrow\;$ match between i and k is scheduled in week w.

  This formal definition of $y_{i,k,w}$ has been adopted from the reference study from Doornmalen et. al. [2].

- **Period assignment (slot).** For each $\{i, k\}$ with $i < k$, week w, and period p:

  $A_{i,k,w,p} \in \{0, 1\}, \quad A_{i,k,w,p} = 1 \;\Leftrightarrow\;$ match $\{i, k\}$ in week w is assigned to period p.

- **Home/away orientation.** For each ordered pair (h, a) with $h \neq a$ and each week w:

  $H_{h,a,w} \in \{0, 1\}, \quad H_{h,a,w} = 1 \;\; \Leftrightarrow \;\;$ team h plays at home against a in week w.

5.2 Objective function

The objective function enforces balance between home and away matches. For each team $t \in T$, the number of home and away games is defined as

$$\text{HomeGames}_t = \sum_{\substack{a \in T \\ a \neq t}} \sum_{w \in W} H_{t,a,w}, \qquad \text{AwayGames}_t = \sum_{\substack{h \in T \\ h \neq t}} \sum_{w \in W} H_{h,t,w}.$$

The imbalance of team t is given by the absolute difference between the two:

$$\text{Imbalance}_t = \left| \text{HomeGames}_t - \text{AwayGames}_t \right|.$$

To control fairness globally, we introduce the maximum imbalance across all teams,

$$\text{MaxImbalance} = \max_{t \in T} \ \text{Imbalance}_t,$$

and set the optimization objective to

$$\min \ \text{MaxImbalance}.$$

5.3 Constraints

To simplify notation, we introduce the set of unordered pairs of teams:

$$\texttt{MATCHES} := \{(i, k) \in T \times T : i < k\}.$$

Each element of `MATCHES` represents a unique match between two distinct teams. All formalizations refer back to the Section 1.4.

5.3.1 Base constraints

1. **Each pair of teams plays exactly once:**

$$\sum_{w \in W} y_{i,k,w} = 1, \quad \forall (i, k) \in \texttt{MATCHES}.$$

2. **Each team plays exactly one match per week:**

$$\sum_{\substack{k \in T \\ k < t}} y_{k,t,w} \; + \; \sum_{\substack{k \in T \\ k > t}} y_{t,k,w} = 1, \quad \forall t \in T, \; \forall w \in W.$$

3. **Each team appears in the same period at most twice:**

$$\sum_{w \in W} \left(\sum_{\substack{k \in T \\ k < t}} A_{k,t,w,p} \; + \; \sum_{\substack{k \in T \\ k > t}} A_{t,k,w,p} \right) \leq 2, \quad \forall t \in T, \; \forall p \in P.$$

5.3.2 Implied constraints

We define as follow the implied constraints described in Section 1.4.1:

1. **Period consistency:** If a match (i, k) is scheduled in week w, it must be assigned to exactly one period in that week.

$$\sum_{p \in P} A_{i,k,w,p} = y_{i,k,w}, \quad \forall (i,k) \in \texttt{MATCHES},\ \forall w \in W.$$

2. **Two teams per period:** Each period in each week must host exactly one match (hence exactly two teams).

$$\sum_{(i,k) \in \texttt{MATCHES}} A_{i,k,w,p} = 1, \quad \forall w \in W,\ \forall p \in P.$$

3. **Symmetry of matches:** If teams i and k play in week w, exactly one of them must be assigned as home and the other as away.

$$H_{i,k,w} + H_{k,i,w} = y_{i,k,w}, \quad \forall (i,k) \in \texttt{MATCHES},\ \forall w \in W.$$

5.3.3 Symmetry breaking constraints

We exploit the symmetry breaking constraints defined in Section 1.4.2.

1. **Home-away symmetry:**

$$\sum_{p \in P} A_{1,2,1,p} = 1.$$

2. **Team permutation symmetry:**

$$y_{\min(1,w+1),\max(1,w+1),w} = 1, \quad \forall w \in W :\ w + 1 \leq n.$$

5.4 Validation

5.4.1 Experimental Design

We adopt the same experimental setup as in the CP model (Section 2.4.1). The formulation was implemented in a solver-independent modeling language (AMPL), and two solvers were considered: Gurobi and CPLEX. Experiments were conducted both with and without the inclusion of symmetry breaking constraints.

5.4.2 Experimental Results

The experimental results are summarized in the following table (as in Section 2.4.2):

ID	Gurobi w/out SB	Gurobi + SB	CPLEX w/out SB	CPLEX + SB
6	0	0	0	0
6 (opt)	**1**	**1**	**1**	**1**
8	0	0	0	0
8 (opt)	**1**	**1**	**1**	**1**
10	0	0	0	0
10 (opt)	**1**	**1**	**1**	**1**
12	0	0	4	4
12 (opt)	**1**	**1**	**1**	**1**
14	16	22	22	18
14 (opt)	**1**	**1**	**1**	**1**
16	79	115	N/A	86
16 (opt)	N/A	**1**	**1**	N/A

Table 7: Results for MIP instances with and without symmetry breaking (SB).

The results demonstrate comparable performance between Gurobi and CPLEX on the tested MIP instances. Both solvers efficiently found and verified optimal solutions for instances up to size 14.
The impact of symmetry-breaking (SB) constraints was not decisive for smaller instances and occasionally increased solve time. However, SB proved critical for the larger, more complex size-16 instance: it enabled CPLEX to find a feasible solution and allowed Gurobi to conclusively prove optimality.

6 Conclusions

We compared CP, SAT, SMT, and MIP approaches to the STS problem. The models were able to solve instances up to 16 teams. Among the tested methods, the SAT formulation achieved the highest limit, successfully solving instances with up to 18 teams. However, the Constraint Programming approach generally delivered the best overall performance. This result is mainly due to the use of advanced search heuristics, such as domain over weighted degree combined with value ordering strategies, which allow CP solvers to efficiently prune the search space and converge more quickly to optimal solutions.

Authenticity and Author Contribution Statement

We declare that the described work is our own and that we did not copy it from someone else. All ideas taken from other sources are properly cited.

AI tools (ChatGPT and DeepSeek) were used to assist in the formal drafting of the report and in defining methods for retrieving and processing the results of the models. They were also used as support for the DIMACS translation and handling.

Author Contributions: Alessia Bedeschi contributed to the SAT and SMT modeling and validation. Julian Pajo contributed to the CP and MIP modeling and validation, and engineered the project's infrastructure, including the Docker containerization and overall project architecture.

References

[1] J.-C. Régin, *Minimization of the Number of Breaks in Sports Scheduling Problems using Constraint Programming*, 2002.

[2] M. J. van Doornmalen, C. Hojny, R. Lambers, & F. C. R. Spieksma, *Integer programming models for round robin tournaments*, European Journal of Operational Research, 310(1), 24–33, 2023.