# Sport Tournament Scheduling

Angela, Castaldo - angela.castaldo2@studio.unibo.it
Jeanritah Roenah Birungi - jeanritah.birungi@studio.unibo.it
Esther, Giuliano - esther.giuliano@studio.unibo.it
Ashmi, Prasad - ashmi.prasad@studio.unibo.it

November 2025

## 1 Introduction

This project addresses the Single Round Robin (SRR) scheduling problem, where each team plays with every other team only once under specified structural and operational constraints [1].

### 1.1 Model Formalisation

#### 1.1.1 Instance Variables

The instance variables listed below specify the common structural elements for all of our model formulations:

- $n$: the number of participating teams (given as input);
- $weeks$: the total number of weeks in the tournament;
- $periods$: the number of matches that can be played in each week;
- $slots$: we fix $slots = 2$, representing the home team (slot 1) and the away team (slot 2);
- $Teams = \{1, \ldots, n\}$: the set of all teams;
- $Weeks = \{1, \ldots, n-1\}$: the set of all weeks in the tournament;
- $Periods = \{1, \ldots, n/2\}$: the set of matches within each week;
- $Slots = \{1, 2\}$: the set of the two slots available in each match.

#### 1.1.2 Objective Function

The objective function focuses on minimizing the difference between the number of home and away matches for each team:

$$\texttt{minimize}\Big( \max_{i \in Team} |h_i - a_i| \Big).$$

#### 1.1.3 Decision Variables

Given $w \in Weeks$, $p \in Periods$, $s \in Slots$, the following decision variable was defined: $team_{p,w,s}$.

#### 1.1.4 Constraints

The problem states the following three constraints that are common across all models:

**C1** the number of teams $n$ must be even: $n \mod 2 = 0$;

**C2** every team plays with every other team only once:
$\forall w \in Weeks,\ \forall (p,s) \neq (p',s'), \quad team_{p,w,s} \neq team_{p',w,s'}$;

**C3** every team plays once a week: $\forall w \in Weeks,\ \forall i \in Teams,$
$\{(p,s) \mid p \in Periods,\ s \in Slots,\ team_{p,w,s} = i\} = 1$;

**C4** every team plays at most twice in the same period over the tournament:
$\forall p \in Periods,\ \forall i \in 1, \ldots, n,$

$$0 \leq \{(w,s) \mid w \in Weeks,\ s \in Slots,\ team_{p,w,s} = i\} \leq 2.$$

### 1.2 Validation

#### 1.2.1 Experimental design

All experiments were executed inside a Docker container based on the python:3.11-slim image. Each experiment was executed using a single thread and a time limit of 300 seconds (5 minutes) per instance, on a host machine equipped with an Intel Core i7-1165G7 CPU (4 cores, 8 threads).

## 2 CP Model

### 2.1 Decision Variables

Given $w \in Weeks$, $p \in Periods$, $s \in Slots$, and $i \in Teams$, the decision variables are:

- $srr_{p,w,s} \in Teams$: contains the fixed round-robin schedule generated following the method shown in [2];
- $perm_{w,p} \in Periods$: is a permutation array of periods for each week $w$;
- $team_{p,w,s} \in Teams$: team assigned to slot $s$ in period $p$ of week $w$, this variable contains the final schedule.

Additional variables are needed for the optimization version of the problem (see Section 1.1.2):

- $swap\_team_{p,w} \in \{0,1\}$: 1 iff the teams in game $(p,w)$ need to be swapped (home/away) in order to obtain a balanced SRR;
- $home\_count_i \in \{0,\dots,n-1\}$: the number of home games of a team $i$;
- $deviation_i = |2 \cdot homeCount_i - (n-1)|$: home/away imbalance of team $i$;
- $max\_deviation \in \{0,\dots,n-1\}$: maximum imbalance over all teams.

## 2.2 Objective Function

In an SRR tournament with an even number of teams, each team plays an odd number of games $(n-1)$, so perfectly balancing home and away games is impossible. Ideally, each team has $\frac{n-1}{2}$ away games and one extra home game. Since MiniZinc doesn't support float arithmetic, the counts are scaled by 2, and the deviation from the ideal imbalance is expressed as: $\big|2{\cdot}\text{home\_count}_i - (n-1)\big|$. The objective *max_deviation* is the maximum deviation across all teams:

$$\text{max\_deviation} = \max(\big|2 \cdot \text{home\_count}_i - (n-1)\big|),$$

so minimizing it reduces the worst-case imbalance. The variable bounds expressed by defining the variables domain are: $home\_count_i \in [1, n-1]$ and $max\_deviation \in [1, n-1]$.

## 2.3 Constraints

The base model enforces all constraints defined in Section 1.1.4. In particular, **C3** and **C4** were implemented as global constraints:

**C3** $\forall w \in \text{Weeks}$, `alldifferent`$\big(\text{team}[p,w,s] \mid p \in \text{Periods}, s \in \text{Slots}\big)$;

**C4** $\forall p \in \text{Periods}$, `global_cardinality_low_up`$\big([\,\text{team}_{p,w,s} \mid w \in Weeks, s \in \text{Slots}\,],\ [\,t \mid t \in \text{Teams}\,],\ [\,0 \mid t \in \text{Teams}\,],\ [\,2 \mid t \in \text{Teams}\,]\big)$.

### 2.3.1 Symmetry Breaking Constraints

The initial model contained many symmetries that greatly expanded the search space. To reduce it, a series of symmetry-breaking constraints were introduced. First we fixed the weekly games using a round-robin schedule, then we assigned a period to each game.

**C5** Fixed round-robin schedule:

**C5.1** Fixing the set of games for the first week (see [2]) as:

$$\text{srr}_{1,p,1} = \begin{cases} \langle 1, 2\rangle & \text{if } p = 1, \\ \langle p+1, n-p+2\rangle & \text{if } p > 1, \end{cases} \quad p \in Periods$$

**C5.2** Assigning subsequent weeks to games using the following formula from [2]: with $p \in Periods,\ w \in Weeks$

$$\mathrm{srr}_{p,w,1} = \begin{cases} 1 & \text{if } \mathrm{srr}_{p,w-1,1} = 1, \\ 2 & \text{if } \mathrm{srr}_{p,w-1,1} = n, \\ \mathrm{srr}_{p,w-1,1} + 1 & \text{otherwise,} \end{cases}$$

with $p \in Periods,\ w \in 2..weeks$

$$\mathrm{srr}_{p,w,2} = \begin{cases} 2 & \text{if } \mathrm{srr}_{p,w-1,2} = n, \\ \mathrm{srr}_{p,w-1,2} + 1 & \text{otherwise,} \end{cases}$$

**C6** Each period from $srr_{p,w,s}$ is mapped to a single period in $team_{p,w,s}$:

$$\texttt{alldifferent}([perm_w | w \in Weeks]);$$

**C7** Breaking period symmetries by fixing the period order for the first week:

$$\forall\ p \in Periods, perm_{1,p} = p;$$

**C8** Channelling constraints linking *srr*, *perm*, *team*, which channels the canonical schedule *srr* through the permutation *perm* to the final team assignment *team*: $\forall w \in \text{Weeks}, \forall p \in \text{Periods}$,

$$\mathrm{team}_{\mathrm{perm}_{w,p},w,1} = \mathrm{srr}_{p,w,1} \wedge \mathrm{team}_{\mathrm{perm}_{w,p},w,2} = \mathrm{srr}_{p,w,2}.$$

## 2.4 Validation

### Experimental Design

Two models were built for each problem variant, decision and optimization, with and without symmetry breaking. Four search strategies were evaluated – **base** (default), **ff** (first fail), **DWD+min** (d_w_deg with indomain min), and **DWD+rand** (dom_w_deg with indomain random) – using the Gecode and Chuffed solvers.

**Experimental results**

With the **Gecode** solver, the decision version performed best running the symmetry breaking model with a custom first-fail indomain min strategy.

| Teams | gecode+sb | | | | gecode+!sb | | | |
|---|---|---|---|---|---|---|---|---|
| | base | ff | DWD+min | DWD+rand | base | ff | DWD+min | DWD+rand |
| 10 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 12 | 0 | 0 | 2 | 1 | N/A | N/A | 19 | 107 |
| 14 | 70 | 44 | N/A | 119 | N/A | N/A | N/A | N/A |
| 16 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| 18 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |

Table 1: Results for the decision version of the problem with Gecode solver

As shown in Table 2, the optimization version of the problem is less efficient (no solution for $n = 16$, since the solver must iterate over multiple solutions until the best one is found, increasing runtime compared to the decision version of the problem.

| Teams | gecode+sb | | | | gecode+!sb | | | |
|---|---|---|---|---|---|---|---|---|
| | base | ff | DWD+min | DWD+rand | base | ff | DWD+min | DWD+rand |
| 8 | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| 10 | **3** | N/A | **1** | N/A | **1** | **1** | N/A | **1** |
| 12 | **3** | N/A | N/A | N/A | N/A | **1** | N/A | N/A |
| 14 | **11** | N/A | N/A | N/A | N/A | N/A | N/A | N/A |

Table 2: Results for the optimization version of the problem with Gecode solver

With **Chuffed**, the decision problem could be solved up to 18 teams, with the default (base) search strategy performing best.

| Teams | chuffed+sb | | | chuffed+!sb | | |
|---|---|---|---|---|---|---|
| | base | ff | DWD+min | base | ff | DWD+min |
| 10 | 0 | 0 | 0 | 0 | 3 | 35 |
| 12 | 0 | 0 | 0 | 44 | 13 | N/A |
| 14 | 1 | 10 | 6 | 13 | N/A | N/A |
| 16 | 12 | 17 | 35 | N/A | N/A | N/A |
| 18 | 199 | N/A | N/A | N/A | N/A | N/A |

Table 3: Results for the decision version of the problem with Chuffed solver

The optimization problem also performed slightly better than in Gecode, yielding balanced schedules up to 12 teams, with symmetry breaking and non-symmetry breaking models performing similarly.

In most cases, models with symmetry-breaking constraints outperformed those without, as those constraints reduce the search space.

| Teams | chuffed+sb | | | chuffed+!sb | | |
|---|---|---|---|---|---|---|
| | base | ff | DWD+min | base | ff | DWD+min |
| 8 | **1** | **1** | **1** | **1** | **1** | **1** |
| 10 | **1** | **1** | **1** | **1** | **1** | **1** |
| 12 | **1** | **1** | N/A | **11** | **1** | N/A |
| 14 | N/A | N/A | N/A | N/A | N/A | N/A |

Table 4: Results for the optimization version of the problem with Chuffed solver

# 3 SAT Model

## 3.1 Decision variables

To encode the Sports Tournament Scheduling (STS) problem as a SAT model, two types of Boolean decision variables were used. Given $w \in Weeks$, $p \in Periods$, and $i, j \in Teams$:

- $home_{i,j,w} \in \{True, False\}$: true iff team $i$ plays at home against team $j$ in week $w$, false otherwise. This variable encodes the home/away relationship between teams.
- $per_{i,w,p} \in \{True, False\}$: true iff team $i$ plays in period $p$ during week $w$, false otherwise. It encodes the period in which $i$ plays.

## 3.2 Objective function

In the optimization case, solving is performed in two stages. First, the full constraint set is checked for feasibility, and, if the provided schedule is optimal, the process ends. If that is not the case, a binary search loop is used. Additional constraints avoid the objective not to exceed a given bound, and Z3 is invoked with increasingly tighter bounds, with unsatisfiable bounds being discarded.

## 3.3 Constraints

The following constraints, already introduced in the previous section, were used in the model:

**C2** : Each pair of teams plays exactly once during the tournament:

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

**C3** : Each team plays exactly one match per week:

$$\forall i \in T, \forall W \in w : \sum_{j \in T, j \neq i} (home_{i,j,w} \bigvee home_{j,i,w}) = 1;$$

**C4** : Each team may appear in the same period at most twice:

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

In addition, the following constraints were included:

**C12** : No team can play against itself:

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

**C13** : Teams that play against each other must play in the same period:

$$\forall i, j \in T, i \neq j, \forall w \in W, \forall p \in P : home_{i,j,w} \implies per_{i,w,p} = per_{j,w,p};$$

**C14** : Mutual exclusion between home and away assignments:

$$\forall i, j \in T, i \neq j, \forall w \in W : \neg home_{i,j,w} \bigvee \neg home_{j,iw};$$

**C15** : Each team is assigned to exactly one period per week:

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

**C16** : Each period contains exactly two teams per week:

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

#### 3.3.1 Symmetry-Breaking Constraints

**C5.1** : Fixing first week pairings: for all pairs $(i, j)$ satisfying $j = n + 1 - i, i = 1, 2, \frac{n}{2}$ it is enforced: $home_{i,j,1} = True, home_{j,i,1} = False$.

#### 3.3.2 Optimization Constraints

Binary search introduces the temporary constraint:

**C17** : $\forall i \in T : -M < diff_i < M$ where $diff_i$ is $|h_i - a_i|$, and $M$ is gradually reduced.

### 3.4 Validation

#### 3.4.1 Experimental design

The models were implemented with the Z3 SMT solver, and optimization variants used a manual binary-search procedure with repeated SAT checks. Each team size n was tested independently.

### 3.4.2 Experimental Results

The tables below allow to compare the four models:

- For the decision model, all instances are solved in milliseconds up to $n = 10$, while starting from $n = 12$ it is possible to observe the effect of symmetry breaking, especially at $n = 16$, where solving time is reduced by about 150 seconds;
- For the optimization model, all feasible cases up to $n = 14$ are solved by both model variants, while at $n = 16$ the non-SB model fails, while the SB variant finds an optimal solution;
- None of the four models was able to find a solution in 300 seconds for $n = 18$ and, as expected, both models report UNSAT for $n = 4$.

| Teams | z3+sb default | z3+!sb default |
|---|---|---|
| 2 | 0 | 0 |
| 4 | UNSAT | UNSAT |
| 6 | 0 | 0 |
| 8 | 0 | 0 |
| 10 | 0 | 0 |
| 12 | 3 | 4 |
| 14 | 35 | 30 |
| 16 | 90 | 240 |
| 18 | N/A | N/A |

Table 5: Results for z3 on model SAT (decision)

| Teams | z3+sb default | z3+!sb default |
|---|---|---|
| 2 | **1** | **1** |
| 4 | UNSAT | UNSAT |
| 6 | **1** | **1** |
| 8 | **1** | **1** |
| 10 | **1** | **1** |
| 12 | **1** | **1** |
| 14 | **1** | **1** |
| 16 | **1** | N/A |
| 18 | N/A | N/A |

Table 6: Results for z3 on model SAT (optimization)

# 4 SMT

The SMT approach models the entire tournament using integer variables and logical constraints. A Z3 solver is used to determine whether a valid schedule exists (decision model) and to additionally minimize the home/away imbalance (optimization model).

## 4.1 Decision Variables

- $T[p][w] = (\text{home}, \text{away})$ where
  - $p \in \{0, \ldots, \text{Periods} - 1\}$
  - $w \in \{0, \ldots, \text{Weeks} - 1\}$

## 4.2 Objective Function

The optimization model extends the decision model with an objective function that minimizes the home/away imbalance. Let $h_i$ be the number of home matches for team $i$, we express the imbalance between home and away teams as: $D_i = |2h_i - (n-1)|$; To force home/away distribution to be as balanced as possible, we then minimize: $D^* = \max_i D_i$

## 4.3 Constraints

**C2** Every pair plays once. Each match is encoded as an integer, so that all *code*s are distinct: $\text{code} = \text{home} \cdot n + \text{away}$;

**C3** Weekly Constraint: one game per week per team: for each week,

$$\text{Distinct}(\{\text{home}, \text{away}\}_{p \in \text{periods}});$$

**C4** Period Constraint: max two appearances of the same team in the same period:

$$\sum_{w} 1(\text{team appears in period } p) \leq 2.$$

### 4.3.1 Symmetry Breaking Constraints

**C9** We enforce home away team ordering inside a single match:

$T_{p,w,\text{home}} < T_{p,w,\text{away}}$. This reduces equivalent symmetric solutions;

**C10** Fixing the first match to reduce simmetry in the solution space:

$$T_{0,0} = (1, 2);$$

**C11** Constraining the decision variables domain:

$$1 \leq T_{p,w,\text{home}}, T_{p,w,\text{away}} \leq n;$$

**C12** Avoiding self matches:

$$T_{p,w,\text{home}} \neq T_{p,w,\text{away}}.$$

## 4.4 Validation

### 4.4.1 Experimental Design

Like SAT, SMT experiments were run using the Z3 solver.

### 4.4.2 Experimental Results

| Teams | z3+sb default | z3+!sb default |
|---|---|---|
| 2 | 0 | 0 |
| 4 | UNSAT | UNSAT |
| 6 | 0 | 0 |
| 8 | 0 | 0 |
| 10 | 54 | 251 |
| 12 | N/A | N/A |

Table 7: Results for the decision version of the problem with Z3 solver

For the decision version of the problem (see Table 7) the solver finds valid schedules for $n = 6, 8, 10$. After $n = 12$, the problem becomes too large and hits the 300-second timeout.

| Teams | z3+sb default | z3+!sb default |
|---|---|---|
| 2 | **1** | **1** |
| 4 | UNSAT | UNSAT |
| 6 | **5** | **1** |
| 8 | **7** | **1** |
| 10 | N/A | N/A |
| 12 | N/A | N/A |

Table 8: Results for the optimization version of the problem with Z3 solver

As we can observe in Table 8 optimization is always harder to solve than feasibility, as to use the objective function we add additional variables and constraints, thus increasing solver complexity. For $n = 6$ and $n = 8$, the solver finds an optimal imbalance. However, the optimization version of $n = 10$ and $n = 12$, cannot be solved by Z3.

Symmetry breaking makes the solver more efficient for the decision version of the problem, however, it makes impossible to find the best balanced schedule in the optimization version of the problem.

# 5 MIP Model

## 5.1 Decision Variables

The MIP formulation uses the following main decision variable:

- $x_{w,p,i,j} \in \{0,1\}$:

$$x_{w,p,i,j} = \begin{cases} 1 & \text{if team } i \text{ hosts team } j \text{ in week } w \text{ and period } p, \\ 0 & \text{otherwise} \end{cases}$$

  Here, $w \in Weeks$, $p \in Periods$, $i, j \in Teams$, $i \neq j$.

Additionally, when optimizing the home/away balance, we define:

- $home\_games[i]$, $away\_games[i]$: integer variables counting home and away games for team $i$;
- $home\_away\_diff[i] \geq 0$: deviation of home minus away games for team $i$;
- $max\_deviation \geq 0$: maximum deviation across all teams, which is the optimization objective.

## 5.2 Objective Function

The optimization objective function is

$$\texttt{minimize } max\_deviation = \max_{i \in Teams} |home\_games[i] - away\_games[i]|.$$

For feasibility-only runs, a dummy objective function equal to 0 is used.

## 5.3 Constraints

The linearized constraints **C1**–**C4** enforce the tournament rules:

**C1** (No self-match): $x_{w,p,i,i} = 0$;

**C2** (Each team plays all others once): $\sum_{w,p}(x_{w,p,i,j} + x_{w,p,j,i}) = 1$;

**C3** (One match per team per week): $\sum_{p,j}(x_{w,p,i,j} + x_{w,p,j,i}) = 1$;

**C4** (At most two matches per period per team): $\sum_{w,j}(x_{w,p,i,j} + x_{w,p,j,i}) \leq 2$.

**Symmetry-breaking Constraints**

Optional constraints the redundant search space and improve solver performance:

**C5** (First-match fixing): $x_{1,1,1,2} = 1$;

**C6** (Lexicographical ordering): enforce lexicographical ordering of weeks

$$\sum_{p,i,j:i\neq j} game\_value[i,j]x_{w,p,i,j} \leq \sum_{p,i,j:i\neq j} game\_value[i,j]x_{w+1,p,i,j},$$

$\forall w < |Weeks|$, with $game\_value[i,j] = (i-1)\cdot n + j$.

The MIP variants correspond to combinations of these constraints:

- `!sb_!lex`: no symmetry breaking;
- `sb_!lex`: first-match fixing only;
- `!sb_lex`: lexicographical ordering only;
- `sb_lex`: both first-match fixing and lexicographical ordering.

## 5.4 Validation

### 5.4.1 Experimental Design

Experiments were run in the Docker container described in Section 1.1.4, using a single thread and a 300-second time limit per instance. Configurations included:

- **Solvers:** Gurobi, CPLEX;
- **Algorithms:** default, primal simplex (`psmplx`), dual simplex (`dsmplx`), barrier (`barr`);
- **Objective:** MaxDeviation or feasibility-only;
- **Models:** !sb_!lex, sb_!lex, !sb_lex, sb_lex.

### 5.4.2 Experimental Results

**Decision Version**

| Teams | gurobi+lex+sb | | | | gurobi+lex+!sb | | | | gurobi+!lex+sb | | | | gurobi+!lex+!sb | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 1 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 1 | 28 | 1 | 12 | 17 | 22 | 17 | 3 | 0 | 1 | 0 | 5 | 1 | 2 | 1 | 2 |
| 12 | 44 | 49 | 43 | 49 | 87 | 3 | 87 | 46 | 13 | 6 | 13 | 4 | 12 | 11 | 12 | 9 |
| 14 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 198 | 285 | 196 | N/A | 58 | 98 | 56 | 21 |
| 16 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 120 | N/A | N/A | 69 | N/A | 68 | N/A |

Table 9: Results for Gurobi on model MIP (decision, no objective)

| Teams | cplex+lex+sb | | | | cplex+lex+!sb | | | | cplex+!lex+sb | | | | cplex+!lex+!sb | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT |
| 6 | 0 | 0 | 0 | 0 | 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 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 1 | 1 | 1 | 1 | 2 |
| 12 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 68 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| 14 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 72 | 73 | 73 | N/A | N/A | N/A | N/A | N/A |

Table 10: Results for CPLEX on model MIP (decision, no objective)

For feasibility-only runs, small instances (n = 2, 6) are solved instantly by all solvers, algorithms, and SB/Lex configurations, while n = 4 is consistently UNSAT. Medium instances (n = 8–12) show sensitivity to solver, algorithm, and symmetry/lexicographic settings: Gurobi benefits from SB and certain algorithm choices, whereas CPLEX is more uniform but times out in some larger-medium configurations. Large instances (n $\geq$ 14) are mostly unsolved, with only a few Gurobi configurations completing for n = 14 and n = 16. Overall, SB and careful algorithm selection are critical for efficiently solving decision-only runs beyond trivial sizes.

### Optimization Version

| Teams | gurobi+lex+sb | | | | gurobi+lex+!sb | | | | gurobi+!lex+sb | | | | gurobi+!lex+!sb | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr |
| 2 | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| 4 | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT |
| 6 | **5** | **1** | **5** | **3** | **5** | **1** | **5** | **3** | **5** | **5** | **5** | **5** | **5** | **1** | **5** | **3** |
| 8 | **3** | **7** | **3** | **5** | **5** | **3** | **5** | **5** | **3** | **4** | **3** | **5** | **5** | **4** | **5** | **5** |
| 10 | **5** | **7** | **5** | **7** | **7** | **6** | **7** | **9** | **5** | **6** | **5** | **7** | **7** | **5** | **7** | **5** |
| 12 | **6** | **7** | **6** | **7** | **9** | **5** | **9** | **9** | **7** | **7** | **7** | **5** | **6** | **8** | **6** | **6** |
| 14 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | **7** | **10** | **7** | **9** | **8** | **11** | **8** | N/A |
| 16 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | **9** | N/A | **9** | **6** | N/A | N/A | N/A | N/A |

Table 11: Results for Gurobi on model MIP (optimization, with objective)

| Teams | cplex+lex+sb | | | | cplex+lex+!sb | | | | cplex+!lex+sb | | | | cplex+!lex+!sb | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr | default | psmplx | dsmplx | barr |
| 2 | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| 4 | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT | UNSAT |
| 6 | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** | **3** |
| 8 | **3** | **3** | **3** | **3** | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **3** | **3** | **3** | **3** |
| 10 | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **7** | **5** | **5** | **5** | **5** | **7** | **7** | **7** | **5** |
| 12 | N/A | N/A | N/A | **7** | **5** | **5** | **5** | **9** | **7** | **7** | **7** | **9** | **5** | **5** | **5** | **5** |
| 14 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | **5** | **5** | **5** | N/A | N/A | N/A | N/A | N/A |

Table 12: Results for CPLEX on model MIP (optimization, with objective)

The MaxDeviation objective slightly increases complexity but also enables pruning, allowing some medium instances (n = 6–12) to be solved faster than decision-only runs. SB consistently improves runtime, while Lex has mixed effects depending on the solver and algorithm. Small instances remain trivial, and large instances (n $\geq$ 14) are generally intractable, with only a few Gurobi configurations completing n = 14 and 16. Overall, solver, algorithm, and SB/Lex combinations strongly influence scalability. In addition, objective-driven pruning provides noticeable benefits for medium instances.

## 5.5 Conclusion

In this project we compared the performance of four different approaches: CP (Constraint Programming), SAT (Satisfiability Problem), SMT(Satisfiability Modulo Theory) and MIP (Mixed Integer Linear Programming).

No model could solve the instance $n = 4$, as the combination of constraints C2, C3 and C4 cannot be satisfied simultaneously. The comparison highlighted how the different approaches can be tuned in order to get the best results:

- **CP** approach got great results for larger instances, with the use of the chuffed solver and by carefully tuning the search strategy;
- In **SAT**, round-robin and other symmetry breaking constraints helped the Z3 solver only when they were not too restrictive;
- The **SMT** approach successfully models the full STS constraints and finds schedules for medium-sized instances, by applying symmetry breaking constraints. However, optimization becomes significantly harder, and larger instances exceed the computational time limit;
- Finally, for **MIP**, the most robust solver was Gurobi. Symmetry breaking and careful algorithm selection can significantly improve performance, and including the objective function often prunes even more the search space. Large instances remain mostly intractable, highlighting that MIP is effective for small-to-medium tournaments but requires well-chosen configurations, or potentially hybrid strategies, for larger ones.

In conclusion, we understood the different strengths offered by the different modelling approaches. While CP and SAT show excellent scalability for feasibility, MIP excels in optimization, and SMT provides a powerful middle ground high expressiveness, strong logical reasoning, and suitability for both feasibility and fairness-driven optimization.

Most importantly, the work shows that good modelling with implied and symmetry breaking constraints matters more than the solver itself. Across all approaches, these carefully designed constraints dramatically improved runtime and solution quality.

## Authenticity and Author Contribution Statement

We declare that this work is entirely our own and has not been copied from any other source. AI tools (ChatGPT) were used solely for grammar and wording refinement in the report and the project README.

### Author Contributions:

- Angela Castaldo: SAT modeling and corresponding report sections.
- Jeanritah Roenah Birungi: SMT modeling and corresponding report sections.
- Ashmi Prasad: MIP modeling and corresponding report sections.
- Esther Giuliano: CP modeling and corresponding report section, and coordination of the project and report structure.
- All authors jointly contributed to the conclusion section.

## References


[1] R. V. Rasmussen, R. V. Rasmussen, M. A. Trick, and M. A. Trick, "Round robin scheduling – a survey," *European Journal of Operational Research*, 2008. DOI: 10.1016/j.ejor.2007.05.046.

[2] P. Van Hentenryck, L. Michel, L. Perron, and J. C. Régin, "Constraint programming in opl," en, in *Principles and Practice of Declarative Programming*, G. Nadathur, Ed., Berlin, Heidelberg: Springer, 1999, pp. 98–116, ISBN: 978-3-540-48164-5. DOI: 10.1007/10704567_6.