(M1) SAT - SOLUTION

⚠ in z3 python framework. Init operations take a lot of time. If you do benchmark between differents encodings into the same file. Result could be not correct.

TEAM, HOUSE OR AWAY, PERIOD, WEEK

1 DECISION VARIABLE: $X_{1011}, X_{1111}, \dots, X_{m\,1\,\frac{m}{2}\,m-1}$

2 DOMAIN: $X_{THPW} \in \{0,1\}$

3 CONSTRAINT

1 • every team plays with every other team only once;
2 • every team plays once a week;
3 • every team plays at most twice in the same period over the tournament.

4 ADDED CONSTRAINT

4 Each game has exactly 2 team
5 Each game cannot be played by 2 home or 2 away team
6 Each team in a day play at home or away (no both!)

EXAMPLE:

$\boxed{m=4}$

ONCE ... ...

	W1	W2	W3
TWICE → P1	1-2	1-3	1-4
... → P2	3-4	2-4	2-3

1 PLAY 3 TIMES X (NOT OK!)

- $m-1$ WEEK
- $m/2$ PERIODS
- m TEAMS

⚠ $m=4$ HAS NO FEASIBLE SOLUTION! ⚠

- MODELLING

$T=1$: $P\left[\underbrace{\overbrace{\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}}^{W}}_{\text{HOUSE}} \underbrace{\begin{pmatrix}1&1&1\\0&0&0\end{pmatrix}}_{\text{AWAY}}\right.$

$T=2$: $\begin{pmatrix}1&0&0\\0&1&1\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$

$T=3$: $\begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&0\\1&0&0\end{pmatrix}$

$T=4$: $\begin{pmatrix}0&0&1\\1&1&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$

CONSTRAINTS (IS USED ISTANCES ABOVE TO HELP VISUALIZATION)

1 every team plays with every other team only once;

$T=1$: $P\left[\underbrace{\overbrace{\begin{pmatrix}1&0&0\\0&0&0\end{pmatrix}}^{W}}_{H=0} \begin{pmatrix}0&1&1\\0&0&0\end{pmatrix}\right.$

$T=2$: $\begin{pmatrix}1&0&0\\0&1&1\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$

$T=3$: $\begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&0\\1&0&0\end{pmatrix}$

$T=4$: $\begin{pmatrix}0&0&1\\1&1&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$

$\boxed{\text{AT MOST ONE}} \rightarrow \forall t_1, t_2\ (t_1 \neq t_2) \rightarrow \text{AT MOST ONE}\Big(\big(\overbrace{(X_{t_1 011} \vee X_{t_1 111})}^{W=1\ P=1} \wedge \overbrace{(X_{t_2 011} \vee X_{t_2 111})}^{W=1\ P=1}\big), \big(\overbrace{(X_{t_1 012} \vee X_{t_1 112})}^{W=2\ P=1} \wedge \overbrace{(X_{t_2 01} \vee X_{t_2 112})}^{W=2\ P=1}\big), \dots, \big((X_{t_1 0PW} \vee X_{t_1 1PW}) \wedge (X_{t_2 0PW} \vee X_{t_2 1PW})\big)\Big)$

$\forall t_1, t_2$ → $\binom{m}{2}$ DIFFERENTS AT MOST → $O(m^2)$

AT MOST ONE → EACH AT MOST HAS $|P \cdot W|$ → $\boxed{(A \vee B) \wedge (C \vee D)}$

```python
clauses = []
for t1 in range(0,team):
    for t2 in range(t1+1,team):
            for p in range(0,periods):
                for w in range(0,weeks):
                    t_i = Or(list(vars[t1,:,p,w].flatten()))
                    t_j = Or(list(vars[t2,:,p,w].flatten()))
                    clauses.append(And(t_i , t_j))
            s.add(at_least_one(clauses))
            # print("1 : number of clause in the at_least_one" , len(clauses)) # Dimension check
            clauses = []
```

$\forall t_1, t_2$

$((t_{1H} \vee t_{1A}) \wedge (t_{2H} \vee t_{2A}))$

MUST BE AT MOST_ONE!

② every team plays once a week;

$\boxed{T=1}$ $\boxed{T=2}$ $\boxed{T=3}$ $\boxed{T=4}$

$$p\left[\overbrace{\begin{pmatrix}1&0&0\\0&0&0\end{pmatrix}}^{w}\right. \begin{pmatrix}0&1&1\\0&0&0\end{pmatrix} \quad \begin{pmatrix}1&1&0\\0&1&1\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix} \quad \begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&0\\1&0&0\end{pmatrix} \quad \begin{pmatrix}0&0&1\\1&1&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$$

$\boxed{\text{EXACTLY ONE}}$ $\longrightarrow \forall t \forall w$ EXACTLY ONE $(X_{t01w} , X_{t11w}, \cdots X_{t1pw})$

$|t \cdot w|$ DIFFERENTS EXACTLY ONE

EACH HAS $|2 \cdot P|$ ELEMENTS

```python
# Constraint2 - Every team plays once at week
for t in range(0,team):
    for w in range(0,weeks):
        c = exactly_one( list(vars[t,:,:,w].flatten()) )
        s.add(c)
```

$|2 \cdot P|$ ELEMENTS

---

③ every team plays at most twice in the same period over the tournament.

$\boxed{T=1}$ $\boxed{T=2}$ $\boxed{T=3}$ $\boxed{T=4}$

$$p\left[\underbrace{\overbrace{\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}}^{w}}_{H=0}\right. \underbrace{\begin{pmatrix}1&1&1\\0&0&0\end{pmatrix}}_{t=1} \quad \begin{pmatrix}1&1&0\\0&1&1\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix} \quad \begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&0\\1&0&0\end{pmatrix} \quad \begin{pmatrix}0&0&1\\1&1&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$$

$\boxed{\text{AT MOST } \overset{K}{2}}$ $\longrightarrow \forall t \forall p$ AT MOST 2 $(X_{t0p1}, \dots, X_{t1pw})$

$|P| \cdot |t|$ DIFFERENTS AT MOST

EACH HAS $|2| \cdot |W|$ ELEMENTS

? ENCODING

$\boxed{\text{AT MOST } 2] K} \to \neg(\underbrace{X_{1011} \wedge X_{1012} \wedge X_{1111}}_{\text{ANY COMBINATION OF K+1 VAR}}) \wedge \neg(X_{1011} \wedge X_{1012} \wedge X_{1112}) \wedge \neg(\cdots)$

$O(m^{2+1})$

```python
# Constraint3 - Every team plays at most twice in the same period over the tournament.
for t in range(0,team):
    for p in range(0,periods):
        c = at_most_k( list(vars[t,:,p,:].flatten()) , k = 2 )
        s.add(c)
```

$|2 \cdot W|$ ELEMENTS

---

4 Each game has exactly 2 team

$\boxed{T=1}$ $\qquad\qquad\qquad\qquad$ $\boxed{T=2}$ $\qquad\qquad$ $\boxed{T=3}$ $\qquad\qquad$ $\boxed{T=4}$

$$p\,[\;\overbrace{\begin{pmatrix}1&0&0\\0&0&0\end{pmatrix}}^{w}\begin{pmatrix}0&1&1\\0&0&0\end{pmatrix} \qquad \begin{pmatrix}1&1&0\\0&1&1\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix} \qquad \begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&0\\1&0&0\end{pmatrix} \qquad \begin{pmatrix}0&0&1\\1&1&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$$

$\boxed{\text{EXACTLY TWO}}$ $\rightarrow$

$$\underbrace{\forall p \, \forall w}_{\downarrow} \quad \text{EXACTLY TWO}\Big(\overbrace{(X_{10pw} \vee X_{11pw})}^{\text{ELEM}}, \dots, (X_{m0pw} \vee X_{m1pw})\Big)$$

$|P \cdot W|$ DIFFERENTS EXACTLY TWO

HAS $|M|$ ELEMENTS

```python
# Constraint4.2 - Each game has exactly 2 team
clauses = []
for p in range(0,periods):
    for w in range(0,weeks):
        for t in range(team):
            x = Or(list(vars[t,:,p,w].flatten()))  ] ELEM
            clauses.append(x)
        s.add(exactly_k(clauses,  k = 2))
        # print("4.2 number of clause in the exactly_k: " , len(clauses)) # Dimension check
        clauses = []
```

---

5 Each game cannot be played by 2 home or 2 away team

$\boxed{T=1}$ $\qquad\qquad\qquad\qquad$ $\boxed{T=2}$ $\qquad\qquad$ $\boxed{T=3}$ $\qquad\qquad$ $\boxed{T=4}$

$$p\,[\;\overbrace{\begin{pmatrix}1&0&0\\0&0&0\end{pmatrix}}^{w}\begin{pmatrix}0&1&1\\0&0&0\end{pmatrix} \qquad \begin{pmatrix}1&1&0\\0&1&1\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix} \qquad \begin{pmatrix}0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}0&0&0\\1&0&0\end{pmatrix} \qquad \begin{pmatrix}0&0&1\\1&1&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}$$

$\boxed{\text{AT MOST ONE}}$ $\rightarrow$

$$\forall_F \, \forall_P \, \forall_W \quad \text{AT MOST ONE}\big(X_{1FPW}, \dots, X_{mFPW}\big)$$

$|F \cdot P \cdot W|$ DIFFERENTS AT MOST ONE

$\downarrow$

⚠ DIFFERENT NAME IN HEULE, SEQUENTIAL, ... ENCODINGS

HAS $|E|$ ELEMENTS

```python
# Constraint5 - Each game cannot be played by 2 home-team or 2 away-team
for h in range(home):
    for p in range(0,periods):
        for w in range(0,weeks):
            s.add(at_most_one(list(vars[:,h,p,w].flatten()) , name = f"ff({h}-{p}-{w})"))   # TODO -> FIND THE BEST NAME FOR EACH CATEGORY (NAME IS IMPORTANT TO DON'T HAVE SAME Y)
            if(h == 0 and p == 0 and w == 0 ):     # DEBUG
                print(list(vars[:,h,p,w].flatten())) # DEBUG
                ttt.add(at_most_one(list(vars[:,h,p,w].flatten()) , name = f"T{h-p-w}")) # DEBUG
            # print("4.3 number of clause in the at_most_one : " , len(list(vars[:,h,p,w].flatten())))  # Dimension check
```

6 Each team in a day play at home or away (no both!)

$\boxed{T=1}$

$$p\left[\overbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}^{w} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}\right.$$

$\boxed{T=2}$

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\boxed{T=3}$

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$\boxed{T=4}$

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\boxed{\text{AT MOST ONE}}$ $\longrightarrow$ $\forall t \; \forall p \; \forall w$ ATMOST ONE (X_{t0pw}, X_{t1pw})

$|t \cdot p \cdot w|$ DIFFERENTS AT MOST ONE

2 ELEMENTS

```python
# Constraint6 - Each team in a day play at home or away cannot both
for t in range(team):
    for p in range(periods):
        for w in range(weeks):
            s.add(at_most_one(list(vars[t,:,p,w].flatten()) , name = f"fo({t-p-w})"))
            # print("4.1 number of clause in the at_most_one: " , len(list(vars[t,:,p,w].flatten()))) # Dimension check
```

(M1) SAT - SOLUTION - V2

BENEFITS →
+ HEULE
- SEQUENTIAL
++ BITWISE
• PAIRWISE (THE SAME)

4 CONSTR

Each game has exactly 2 team + Each game cannot be played by 2 home-team or 2 away-team

↓

CONSTRAINTS 4,5,6 OF VERSION1

$\boxed{\text{EXACTLY ONE}} \rightarrow \underbrace{\forall_F \forall_P \forall_W}_{|P \cdot W \cdot F| \text{ DIFFERENTS}} \; \text{EXACTLY ONE} \underbrace{\left(X_{0FPW}, X_{1FPW}, \dots, X_{mFPW}\right)}_{m \text{ ELEMENTS}}$

```python
# Constraint 4 - Each game has exactly 2 team + Each game cannot be played by 2 home-team or 2 away-team
for h in range(home):
    for p in range(periods):
        for w in range(weeks):
            # exactly one home and exactly one away → exactly 2 teams, opposite sides
            heule.add(exactly_one_heule([vars[t,h,p,w] for t in range(team)] , name = f'h{h}p{p}w{w}') )
```

(M1)SAT - SOLUTION - V3

BENEFITS →
- FIRST COLUMN (m) SECOND
- FIRST LINE (m)

→
++ PAIRWISE
-- HEULE
• BITWISE
• SEQUENTIAL

1 SIMMETRY BREAKING

↓

YOU FIXED IN THE FIRST $\boxed{\text{COLUMN}}$ MATCH FROM 1 TO M/2

EX: →

$\boxed{m=6}$

	W1
P1	(1,?)
P2	(2,?)
P3	(3,?)

FIXED COLUMN

(?,1) IS THE SAME

	W1	W2	....
P1	(1,?)	(2,?)	...

⚠ IS ALSO POSSIBLE TO FIX a LINE

↓

$\boxed{\text{CUSTOM ENCODINGS}} \rightarrow (X_{1011} \vee X_{11\underset{\text{LINE}}{1}\underset{\text{COLUMN}}{1}}) \wedge (X_{1012} \vee X_{11\underset{\text{LINE}}{2}\underset{\text{COLUMN}}{1}}) \dots (X_{1011} \vee X_{11\underset{\text{LINE}}{\frac{m}{2}}1})$

Encodings

$O(m^2)$ CLAUSES

- PAIRWISE $\rightarrow$ AT_MOST_ONE $(X_1, X_2, X_3, X_4) \rightarrow (\neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee \neg X_3) \cdots \binom{m}{2}$

- SEQUENTIAL $\rightarrow$ AT_MOST_ONE (X_1, X_2, X_3, X_4)

m CLAUSES + $m-1$ AUXILIAR VAR

$\downarrow$

$(\neg X_1 \vee S_1) \wedge$
$(\neg X_2 \vee S_2) \wedge$
$(\neg X_3 \vee S_3) \wedge$

$(\neg S_1 \vee S_2) \wedge$
$(\neg S_2 \vee S_3) \wedge$

$(\neg X_2 \vee \neg S_1) \wedge$
$(\neg X_3 \vee \neg S_2) \wedge$
$(\neg X_4 \vee \neg S_3)$

$(\neg X_1 \vee S_1) \wedge X \wedge (\neg X_2 \vee \neg S_1)$

(a) If x_i is true, then s_i must be true

- $\neg x_1 \vee s_1$
- $\neg x_2 \vee s_2$
- $\neg x_3 \vee s_3$

(b) Monotonicity: once s_i is true, all following must be true

- $\neg s_1 \vee s_2$
- $\neg s_2 \vee s_3$

(c) At most one: if x_{i+1} is true, then s_i must be false

- $\neg x_2 \vee \neg s_1$
- $\neg x_3 \vee \neg s_2$
- $\neg x_4 \vee \neg s_3$

- ADDITION IN PROPOSITIONAL LOGIC

- 2 DIGITS

a_1	a_0	
b_1	b_0	
c_2	c_1	c_0

$\downarrow$

$d_0 = a_0 \oplus b_0 \oplus c_0$

$c_1 = MAJ(a_0, b_0, c_0)$

$d_1 = a_1 \oplus b_1 \oplus c_1$

$c_2 = MAJ(a_1, b_1, c_1)$

$\xrightarrow{\text{LOGIC}}$

$d_0 \equiv \left(\neg\left((\neg a_0 \wedge b_0) \vee (a_0 \wedge \neg b_0)\right) \wedge c_0\right) \vee \left(\left((\neg a_0 \wedge b_0) \vee (a_0 \wedge \neg b_0)\right) \wedge c_0\right)$

$c_1 \equiv \dots$ MAJ(a_0, b_0, c_0)

...

$\oplus \rightarrow$ XOR $\longrightarrow$

A	B	
0	0	F
0	1	T
1	0	T
1	1	F

$\longrightarrow (\neg A \wedge B) \vee (A \wedge \neg B)$

MAJ $\rightarrow$ MAJORITY FUNCTION (AT LEAST 2 OF 3 TRUE) $\longrightarrow$

A	B	C	
0	0	0	F
0	0	1	F
0	1	0	F
0	1	1	T
1	0	0	F
1	0	1	T
1	1	0	T
1	1	1	T

$\longrightarrow (A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$

(M2) SAT - SOLUTION - V3

1 DECISION VARIABLE:

$\binom{m}{2}$ ALL POSSIBLE TEAM COMBINATION

is t1,t2 play a game on week w ?

$$\underbrace{M[0][1][0], \dots, M[m][m-1]}_{\binom{m}{2}\text{ ALL POSSIBLE TEAM COMBINATION}}[w] \quad \Big\} \binom{m}{2} \cdot |w| \text{ VARIABLES}$$

$$H[0][0], \dots, H[t][w] \ \Big\} |t \cdot w| \text{ VARIABLES}$$

is team t home in week w ?

$$P[0][0][0], \dots, P[t][p][w] \ \Big\} |t \cdot p \cdot w| \text{ VARIABLES}$$

is team is playing in the period and week p,w ?

2 DOMAIN: $M, H, P \in \{0,1\}$

3 CONSTRAINT

1 • every team plays with every other team only once;
2 • every team plays once a week;
3 • every team plays at most twice in the same period over the tournament.

4 ADDED CONSTRAINT

4 • in a game (1 home team - 1 away team)
5 • each game is played by 2 team
- TODO

5 PRECOMPUTING? ✓ YES (ROUND ROBIN PAIRS) → DESCRIPTION ABOVE!

6 OPTIMIZATION ✓ DESCRIPTION ABOVE!

= NUMBER OF CARDINALITY COSTRAINTS FOR ∂ CONSTRAINTS (WHEN YOU USE ENCODINGS LIKE HEULE, SEQUENTIAL, ... YOU SHOULD HAVE m UNIQUE NAMES (LABEL))

= NUMBER OF VARIABLES FOR ∂ CARDINALITY CONSTRAINT

✓ 1 • every team plays with every other team only once;

EXACTLY_ONE → $\forall t_1, t_2$ $\quad E\_ONE(\overbrace{M[t_1][t_2][0], \dots, M[t_1][t_2][w]}^{|W|})$

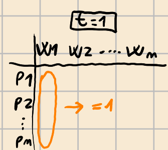
MIP → $\forall t_1, t_2$ $\quad \underbrace{M[t_1][t_2][0] + \dots M[t_1][t_2][w]}_{|W|} = 1$

```
# Constraint 1 : every team plays with every other team only once;
for t1 in range(team):
    for t2 in range(t1 + 1, team):
        S.add(PbEq([(M[t1][t2][w], 1) for w in range(weeks)], 1))
```

✓ 2 • every team plays once a week;

EXACTLY_ONE → $\forall w, t$ $\quad E\_ONE(\overbrace{P[t][0][w], \dots, P[t][0][w]}^{|P|})$

MIP → $\forall w, t$ $\quad \overbrace{P[t][0][w] + \dots + P[t][0][w]}^{|P|} = 1$ →



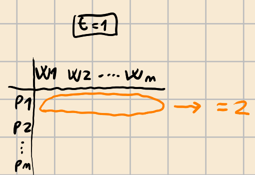

```
# Constraint2 : each team play exactly one per week: sum over p of P[t][p][w] == 1
for w in range(weeks):
    for t in range(team):
        # S.add(PbEq([(P[t][p][w], 1) for p in range(periods)], 1))   # SAT translation of x1+x2+x3 = 1 (PbEq also enabled weight 3x1 + 4x2 + ...)
        S.add(exactly_k([P[t][p][w] for p in range(periods)], 1))   # TODO : use exactly one of sequential,heule,bitwise

```

✓ 3 • every team plays at most twice in the same period over the tournament.

EXACTLY_TWO → $\forall t, p$ $\quad E\_TWO(\overbrace{P[t][p][0], \dots, P[t][p][w]}^{|W|})$

MIP → $\forall t, p$ $\quad \overbrace{P[t][p][0] + \dots + P[t][p][w]}^{|W|} = 2$ →




```
# Constraint3 : each team plays two per period : sum over weeks of P[t][p][w] <= 2
for t in range(team):
    for p in range(periods):
        S.add(PbLe([(P[t][p][w], 1) for w in range(weeks)], 2))
```

4 • in a game (1 home team - 1 away team)

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

$\boxed{\text{CUSTOM}} \rightarrow \forall w, t_1, t_2 \quad M[t_1][t_2][w] \rightarrow XOR(H[t_1][w], H[t_2][w])$

$\boxed{\text{MIP}} \rightarrow ?$

```
# Constraint4 : Home/away consistency: when (t1,t2) plays in w, HOME differs
for w in range(weeks):
    for t1 in range(team):
        for t2 in range(t1 + 1, team):
            S.add(Implies(M[t1][t2][w], Xor(HOME[t1][w], HOME[t2][w])))
```

5 each game is played by 2 team

$\boxed{\text{EXACTLY\_TWO}} \rightarrow \forall p, \forall w \quad E\_TWO(\overbrace{P[0][p][w], \ldots, P[t][p][w]}^{|t|})$

$\boxed{\text{MIP}} \rightarrow \forall p, \forall w \quad \overbrace{P[0][p][w] + \ldots + P[t][p][w]}^{|t|} = 2 \longrightarrow$

GAME(p,w)

$\downarrow$

$t_1 + t_2 + \cdots + t_m = 2$

```
# Constraint5 : Each game is played by 2 team : sum over t of P[t][p][w] == 2
for w in range(weeks):
    for p in range(periods):
        S.add(PbEq([(P[t][p][w], 1) for t in range(team)], 2))     # SAT translation of x1+x2+x3 =2 (PbEq also enabled weight 3x1 + 4x2 + ...)
        # S.add(exactly_k([P[t][p][w] for t in range(team)], 2))     # TODO : use exactly k of sequential
```

LINK - VARIABLES

t1 and t2 MUST PLAY IN THAT WEEK.

$\boxed{\text{CUSTOM}} \rightarrow \forall w, t_1, t_2 \quad M[t_1][t_2][w] \rightarrow \overbrace{(P[t_1][0][w] \land P[t_2][0][w]) \lor \cdots \lor (P[t_1][p][w] \land P[t_2][p][w])}^{|P|}$

$\boxed{\text{MIP}} \rightarrow ???$

```
# B) Link periods to pair variables M
for w in range(weeks):
    for t1 in range(team):
        for t2 in range(t1 + 1, team):
            # If the pair plays in week w, they share exactly one period
            S.add(Implies(M[t1][t2][w], Or([And(P[t1][p][w], P[t2][p][w]) for p in range(periods)])))

            # If two teams share (period p, week w), then that (t1,t2) is the match that week
            for p in range(periods):
                S.add(Implies(And(P[t1][p][w], P[t2][p][w]), M[t1][t2][w]))
```

ROUND ROBIN PAIRS → PAIR GENERATION

```python
def round_robin_pairs(n):
    arr = list(range(n))
    weeks_list = []
    for w in range(n - 1):
        pairs = []
        for i in range(n // 2):
            a = arr[i]
            b = arr[n - 1 - i]
            if a < b: pairs.append((a, b))
            else:     pairs.append((b, a))
        weeks_list.append(pairs)
        # rotate (fix arr[0])
        arr = [arr[0]] + [arr[-1]] + arr[1:-1]
    return weeks_list
```

LAST ELEM OF THE LIST

FROM 2 TO THE LAST (NOT INCLUDED)

$\boxed{EX:\ n=4}$

ARR = [0,1,2,3]

W_L = [ ]

↓ ...

W_L = [[0,3] [1,2]] $\boxed{W=1}$

ARR = [0,3,1,2]

W_L = [..., [0,1] [2,3]] $\boxed{W=2}$

↓ ...

W=1 W=2 W=3

[[(0, 3), (1, 2)], [(0, 2), (1, 3)], [(0, 1), (2, 3)]]

- USING PRECOMPUTED LIST OF ROUND ROBIN

```python
if WARMUP:
    # Pin ONLY the pairings to a valid round-robin schedule to avoid symmetry blow-up.
    rr_weeks = round_robin_pairs(team)  # weeks -> list[(a,b)]
    M_true = {(min(a,b), max(a,b), w) for w, pairs in enumerate(rr_weeks) for (a,b) in pairs}
    for t1 in range(team):
        for t2 in range(t1+1, team):
            for w in range(weeks):
                if (t1, t2, w) in M_true:
                    S.add(M[t1][t2][w])
                else:
                    S.add(Not(M[t1][t2][w]))
```

EXTENDED

```python
M_true = set()  # start with an empty set
    for w, pairs in enumerate(rr_weeks):
        print(w , pairs)
        for a, b in pairs:
            t1 = min(a, b)
            t2 = max(a, b)
            M_true.add((t1, t2, w))
```

↓ EX: $\boxed{n=4}$

RR_WEEKS = [[(0, 3), (1, 2)], [(0, 2), (1, 3)], [(0, 1), (2, 3)]]

OPTIMIZATION

- EXAMPLE (n=6)

$\boxed{\text{H[t][w] TABLES}}$

	W1	W2	W3	W4	W5	TEAM IMBALANCE
T0	0	0	0	1	1	→ 1
T1	0	0	0	0	0	→ 5
T2	1	0	1	0	1	→ 1
T3	1	1	0	1	0	→ 1
T4	0	0	1	0	0	→ 3
T5	1	1	0	1	1	→ 1

AWAY
HOME

TOT: 12 != n => NOT PERFECT

THE PERFECT BALANCES SHOULD BE n USING ATLEAST_K (...)

(n-2)//2

- CONSTRAINT

① AT_LEAST_K (P[0][1], P[0][2], ...) → # AT_LEAST_K HOME MATCH FOR TEAM_0

② AT_LEAST_K (¬ P[0][1] ....) → # AT_LEAST_K AWAY MATCH FOR TEAM_0