A Pragmatic Approach for Solving the Sports Scheduling Problem

Angelos Dimitsas[1], Christos Gogos[1], Christos Valouxis[2], Alexandros Tzallas[1], and Panayiotis Alefragis[3]

[1] University of Ioannina, Dept. of Informatics and Telecommunications, Arta, Greece
{a.dimitsas, cgogos, tzallas}@uoi.gr
[2] University of Patras, Dept. of Electrical and Computer Engineering, Patras, Greece
cvalouxis@upatras.gr
[3] University of Peloponnese, Dept. of Electrical and Computer Engineering, Greece, Patras
alefrag@uop.gr

Abstract. Sports Scheduling is a problem with many variations, regarding the sport type, the various hard rules that have to be obeyed and the quality metrics that are expected to be met. Various stakeholders including organizers, teams, spectators and others have interest in acquiring high quality schedules that satisfy rules and constraints crucial from their point of view. In this work we propose an approach for solving the Sports Scheduling problem as defined in the International Timetabling Competition 2021 (ITC2021). We describe the analytical formulation of each constraint, as it can be modeled for a CP solver and five moves that can be used for altering a schedule by a metaheuristic. We also document the experience gained in trying to address the problem using heuristics, metaheuristics, Constraint Programming and the capable ORTools CP/SAT solver. Despite the computational hardness of the problem instances, our approach managed to achieve good results for most of them.

Keywords: Sports Scheduling· Constraint Programming · Simulated Annealing

1 Introduction

Sports Scheduling is the problem of constructing a tournament schedule consisting of matches among competing teams that form a league. The schedule should satisfy the constraints imposed by the tournament's rules and be 'invisible' in the sense that the various stakeholders such as organizers, teams, spectators, and others should not have legitimate reasons to question it.

Sports scheduling exists for as long as there are sports and teams willing to participate in tournaments with matches against each other. For some sports, like tennis, instead of teams, individual athletes compete. Furthermore, there are tournaments, like chess or other board games tournaments, where the actual matches would be hardly identified as sports, in the typical sense. Esports (electronic sports), is another example of a competition for which its events should be scheduled according to a carefully crafted plan. The same principles regarding scheduling apply to all previously identified cases of tournaments and are special instances of the sports scheduling problem.

Several variations of tournaments exist including single round tournaments, double round tournaments, tournaments with elimination games, compact tournaments (all teams have matches in every timeslot), etc. Some heuristics for constructing sport schedules are known for many years, like the circle [11] method and the Berger [3] method. But when constraints are added the problem quickly becomes very hard to solve. Such constraints might involve the avoidance of consecutive away games for all or some teams, the enforcement of minimum distances (number of time slots) between a match and the rematch, and many others. In this paper, an approach of generating high quality schedules for the compact, double round robin (2RR) type of tournament, is presented. This approach is based on several moves that keep the schedule feasible and a Constraint Programming formulation that results in a model capable of performing complex moves when no progress can be achieved otherwise. A move is eventually a series of changes involving teams participating in a set of matches. Some moves result in better schedules and some others may lay the foundation for performing subsequent moves that will lead to even better schedules.

2 Related work

Several real life tournaments have been addressed using automated techniques involving mathematical programming, constraint programming, metaheuristics and heuristics; e.g., the Belgian soccer league [7], the Brazilian soccer tournament [16], the Finnish national youth ice hockey league [14], the Chilean soccer leagues [1], the South American qualifiers for FIFA 2018 [6], the Icelandic football league [8].

Lewis and Thompson, in [12] present the association of the sports scheduling problem to the a graph coloring problem. Moreover, an edge coloring presentation of the problem is available at [10].

Regarding the exploration of the solution space in [4] it is established that the solution space is not connected by the usually used neighborhood structures, i.e. it's impossible starting from a feasible timetable to reach all other possible timetables just by performing the usual heuristic moves proposed in the bibliography, and [9] proposed a new neighborhood operator to handle this issue.

Since sports timetabling usually results in problems of big sizes, decomposition approaches can be advantageous. In [18] a first schedule then break approach was tried. First it was decided when teams would meet, and the home advantage is decided later. The opposite, first break then schedule approach can be seen at [17], first it is decided where each team plays at home and the teams are paired later. An effort on minimizing breaks is available at [13]. A research on feasible home-away patterns is presented at [2].

3 Problem Description

The problem description can be found at [19] and it refers to tournaments categorized as time-constrained double round robin. Time-constrained or compact means that the timetable uses the minimum number of time slots, i.e. in each time slot all teams play in matches.

3.1 The Base Constraints

The base constraints for each tournament are the format of the tournament. All tournaments are in double round robin format, i.e. each team has two matches against every other team, one at home and one away. Some of the tournaments contain the Phase rule; the timetable is divided in half (two phases), a match and its rematch must be in a different phase. All tournaments are compact.

3.2 The Hard and Soft Constraints of ITC2021

All type of constraints can be either hard or soft as of the type attribute. Hard constraints must be satisfied and soft constraints create deviations penalized in the objective function. There are 9 types of constraints in 5 different constraint categories.

Capacity Constraints Capacity constraints regulate the matches played by a team or a group of teams at home or away.

CA1 constraints regulate the number of matches a team plays at home or away in specific slots.

CA2 constraints regulate the number of matches a team plays at home or away in specific slots against specific teams.

CA3 constraints regulate the number of matches a team plays at home or away in a sequence of slots.

CA4 constraints regulate the number of matches a group of teams play at home or away in specific slots against specific teams.

Game Constraints Game constraints enforce or forbid specific matches in certain slots.

GA1 constraints deal with fixed or forbidden matches to slots assignments.

Break Constraints If a team plays a game with the same home-away status as its previous game, we say it has a break.

BR1 constraints limit the breaks a team has in specific slots.

BR2 constraints limit the breaks a group of teams has in specific slots.

Fairness Constraints Fairness constraints attempt to increase fairness and attractiveness of a tournament.

FA2 constraints limit the difference in played home games of set of teams.

Separation Constraints Separation constraints regulate the number of slots between matches involving the same pairs of teams.

SE1 limits the difference between matches and rematches of the same teams.

4 A Pragmatic Approach

4.1 Heuristic Moves

We have identified five different heuristic moves that create a new timetable from an existing one. The new timetable conforms to the base constraints. In Table 1 a timetable is presented, this timetable will be used as a starting timetable for the examples for all available moves.

Table 1. Double round robin tournament created with the circle method.

1	2	3	4	5	6
1-4	1-3	1-2	4-1	3-1	2-1
2-3	4-2	3-4	3-2	2-4	4-3

- SwapHomes. Two teams $t_1 \neq t_2$ are selected, we swap match $t_1 - t_2$ with $t_2 - t_1$. Two matches are affected. An example can be seen in Table 2.

Table 2. Timetable 1 after SwapHomes move for teams 1 and 4.

1	2	3	4	5	6
4-1	1-3	1-2	**1-4**	3-1	2-1
2-3	4-2	3-4	3-2	2-4	4-3

- SwapRounds. Two slots $s_1 \neq s_2$ are selected, we swap the matches of s_1 with those of s_2. For a tournament of T teams, T matches are affected. An example can be seen in Table 3. For tournaments with the Phase rule swapping slots is allowed only on slots of the same phase.
- Swap Teams. Two teams $t_1 \neq t_2$ are selected, we swap team t_1 with t_2 in all matches. For a tournament of T teams, $4(T-1)$ matches are affected. An example can be seen in Table 4.
- PartialSwapTeams. Two teams $t_1 \neq t_2$ are selected, we swap opponents of team t_1 with those of t_2 in all slots and also keep in mind that if a match is already scheduled to also schedule its rematch. For a tournament of T teams, $4(T-1)-2$ matches are affected. An example can be seen in Table 5.

Table 3. Timetable 1 after SwapRounds move for slots 1 and 5.

1	2	3	4	5	6
3-1	1-3	1-2	4-1	**1-4**	2-1
2-4	4-2	3-4	3-2	**2-3**	4-3

Table 4. Timetable 1 after SwapTeams move for teams 1 and 2.

1	2	3	4	5	6
2-4	**2**-3	**2-1**	4-**2**	3-**2**	**1-2**
1-3	4-**1**	3-4	3-**1**	**1**-4	4-3

– PartialSwapRounds. One match m is selected and moved from slot A to slot B, matches involving teams from m are moved to slot A, an ejection sequence between slots A and B occurs until each team participates in one match per slot. For tournaments with the Phase rule the match m is allowed only to move to a slot in the same phase. The number of matches affected varies. An example can be seen in Table 6.

4.2 Simulated Annealing

Simulated Annealing [21] is a well-known optimization technique that manages to produce near optimal results for a variety of problems. It escapes local minima by accepting inferior solutions with high probability during early stages of the process. This probability diminishes as the process continues. In particular, the acceptance probability of an inferior solution at step k of the procedure is given by $\frac{e^{f(cur)-f(new)}}{T_k}$ where $f(cur)$ is the cost of the current best solution, $f(new)$ is the cost of the new solution and T_k is the temperature after k decreases from an initial temperature T. The temperature is decreased based on the cooling factor a, using the formula $T_k = aT_{k-1}$.

There seems to be some art in calibrating Simulated Annealing to get the best possible results [5]. In our approach, after trial and error the following values were chosen, $T = 5$, $a = 0.999$. The procedure restarts when the temperature of 0.1 is reached.

4.3 Constraint Programming Formulation

Decision Variables For the set of teams $\mathbb{T}$, the set of slots $\mathbb{S}$, with S as the number of available slots and T as the number of teams we define the following binary decision variables.

Table 5. Timetable 1 after PartialSwapTeams move for teams 1 and 3.

1	2	3	4	5	6
1-**2**	1-3	1-**4**	**2**-1	3-1	**4**-1
4-3	4-2	3-**2**	3-**4**	2-4	**2**-3

Table 6. Timetable 1 after placing match 1-2 in slot 5. Note that in small tournaments the effect is always a swap of the slots, but in larger tournaments only some of the matches will exchange slots.

1	2	3	4	5	6
1-4	1-3	**3-1**	4-1	**1-2**	2-1
2-3	4-2	**2-4**	3-2	**3-4**	4-3

$$x_{i,j,s} = \begin{cases} 1, \text{ If team i plays against team j in slot s} \\ 0, \qquad \text{Otherwise} \end{cases} \quad \forall i, j \in \mathbb{T}, i \neq j, \forall s \in \mathbb{S} \tag{1}$$

To monitor the home away pattern we define:

$$y_{i,s} = \begin{cases} 1, \text{ If team i plays at home in slot s} \\ 0, \qquad \text{Otherwise} \end{cases} \quad \forall i \in \mathbb{T}, \forall s \in \mathbb{S} \tag{2}$$

We enforce the home-away pattern to follow the timetable:

$$y_{i,s} = \sum_{j=1}^{T} x_{i,j,s} \quad \forall i \in \mathbb{T}, i \neq j, \forall s \in \mathbb{S} \tag{3}$$

In all instances, constraints regarding breaks do not take into consideration if the breaks occur at Home or Away, so we just have to keep track in which slots a general break occurs:

$$z_{i,s} = \begin{cases} 1, \text{ If team i has a break in slot s} \\ 0, \qquad \text{Otherwise} \end{cases} \quad \forall i \in \mathbb{T}, \forall s \in \mathbb{S} \tag{4}$$

We enforce the break pattern to follow the home-away pattern:

$$z_{i,s} = \begin{cases} 1, \; y_{i,s} = y_{i,s-1}, s > 1 \\ 0, \qquad s = 1 \end{cases} \quad \forall i \in \mathbb{T}, \forall s \in \mathbb{S} \tag{5}$$

Base Constraints Each team must play exactly one match at home against each other team:

$$\sum_{s=1}^{S} x_{i,j,s} = 1 \quad \forall i, j \in \mathbb{T}, i \neq j \tag{6}$$

To satisfy the compactness rule each team plays one match in each slot:

$$\sum_{j=1}^{T}(x_{i,j,s} + x_{j,i,s}) = 1 \quad \forall i \in \mathbb{T}, i \neq j, \forall s \in \mathbb{S} \tag{7}$$

For instances with the phase rule a match and its rematch must be in different phases:

$$\sum_{s}^{S/2}(x_{i,j,s} + x_{j,i,s}) = 1 \quad \forall i, j \in \mathbb{T}, i < j, \forall s \in \mathbb{S} \tag{8}$$

CA1 Constraints Each CA1 constraint with team t_c in "teams" field, with $\mathbb{S}_c$ the set of teams in "slots" field and max_c in "max" field, triggers a d_c deviation.

CA1 with mode="H":

$$d_c = \sum_{s \in \mathbb{S}_c} y_{t_c,s} - max_c \tag{9}$$

CA1 with mode="A" and S_c the size of $\mathbb{S}_c$:

$$d_c = S_c - \sum_{s \in \mathbb{S}_c} y_{t_c,s} - max_c \tag{10}$$

CA2 Constraints Each CA2 with team t_1 in "teams1" field, with $\mathbb{S}_c$ the set of slot in "slots" field, with $\mathbb{T}_c$ the set of slots in "teams2" field, with max_c in "max" field triggers a deviation d_c.

CA2 with mode="H":

$$d_c = \sum_{t_2 \in \mathbb{T}_c} \sum_{s \in \mathbb{S}_c} x_{t_1,t_2,s} - max_c \tag{11}$$

CA2 with mode="A":

$$d_c = \sum_{t_2 \in \mathbb{T}_c} \sum_{s \in \mathbb{S}_c} x_{t_2,t_1,s} - max_c \tag{12}$$

CA2 with mode="HA":

$$d_c = \sum_{t_2 \in \mathbb{T}_c} \sum_{s \in \mathbb{S}_c} (x_{t_1,t_2,s} + x_{t_2,t_1,s}) - max_c \tag{13}$$

CA3 Constraints Each CA3 with $\mathbb{T}_{c1}$ the set of teams in "teams1" field, with $\mathbb{S}_c$ as the slots in "slots" field, with $\mathbb{T}_{c2}$ the set of teams in "teams2" field and max_c in "max" field triggers deviations d_c for each team in $\mathbb{T}_{c1}$ and for all slot sequences $\mathbb{S}_c$ of size $intp$ in "intp" field.

CA3 with mode="H":

$$d_c = \sum_{t_2 \in \mathbb{T}_{c2}} \sum_{s=k}^{k+intp} x_{t_1,t_2,s} - max_c \quad \forall t_1 \in \mathbb{T}_{c1}, t_1 \neq t_2, 1 \leq k \leq S_c - intp \tag{14}$$

CA3 with mode="A":

$$d_c = \sum_{t_2 \in \mathbb{T}_{c2}} \sum_{s=k}^{k+intp} x_{t_2,t_1,s} - max_c \quad \forall t_1 \in \mathbb{T}_{c1}, t_1 \neq t_2, 1 \leq k \leq S_c - intp \quad (15)$$

Special case: In all instances there are at most two Hard CA3 constraints, one with mode="H" and the other with mode="A", $\mathbb{T}_{c1} = \mathbb{T}_{c2} = \mathbb{T}$, $\mathbb{S}_c = \mathbb{S}$, max_c is always 2 and $intp$ is always 3. If both rules exist then the home-away patterns "HHH" and "AAA" cannot appear, so for those instances a team cannot have two breaks in a row:

$$z_{i,s} + z_{i,s-1} \leq 1 \quad \forall i \in \mathbb{T}, \forall s \in \mathbb{S}, s > 2 \quad (16)$$

CA4 Constraints Each CA4 with mode2="GLOBAL" triggers a deviation d_c equal to the sum of the matches between the set of teams $\mathbb{T}_{c1}$ in "teams1" field and the set of teams $\mathbb{T}_{c2}$ in "teams2" field in all slots of the set $\mathbb{S}_c$ in "slots" field over max_c in "max" field.

CA4 with mode2="GLOBAL" and mode1="H":

$$d_c = \sum_{s \in \mathbb{S}_c} \sum_{t_1 \in \mathbb{T}_{c1}} \sum_{t_2 \in \mathbb{T}_{c2}} x_{t_1,t_2,s} - max_c \quad t_1 \neq t_2 \quad (17)$$

CA4 with mode2="GLOBAL" and mode1="A":

$$d_c = \sum_{s \in \mathbb{S}_c} \sum_{t_1 \in \mathbb{T}_{c1}} \sum_{t_2 \in \mathbb{T}_{c2}} x_{t_2,t_1,s} - max_c \quad t_1 \neq t_2 \quad (18)$$

CA4 with mode2="GLOBAL" and mode1="HA":

$$d_c = \sum_{s \in \mathbb{S}_c} \sum_{t_1 \in \mathbb{T}_{c1}} \sum_{t_2 \in \mathbb{T}_{c2}} (x_{t_1,t_2,s} + x_{t_2,t_1,s}) - max_c \quad t_1 \neq t_2 \quad (19)$$

Each CA4 with mode2="EVERY" triggers a deviation d_c for each slot of the slots set $\mathbb{S}_c$ in "slots" field equal to the sum of the matches between the set of teams $\mathbb{T}_{c1}$ in "teams1" field and the set of teams $\mathbb{T}_{c2}$ in "teams2" field over max_c in "max" field.

CA4 with mode2="EVERY" and mode1="H":

$$d_c = \sum_{t_1 \in \mathbb{T}_{c1}} \sum_{t_2 \in \mathbb{T}_{c2}} x_{t_1,t_2,s} - max_c \quad t_1 \neq t_2, \forall s \in \mathbb{S}_c \quad (20)$$

CA4 with mode2="EVERY" and mode1="A":

$$d_c = \sum_{t_1 \in \mathbb{T}_{c1}} \sum_{t_2 \in \mathbb{T}_{c2}} x_{t_2,t_1,s} - max_c \quad t_1 \neq t_2, \forall s \in \mathbb{S}_c \quad (21)$$

CA4 with mode2="EVERY" and mode1="HA":

$$d_c = \sum_{t_1 \in \mathbb{T}_{c1}} \sum_{t_2 \in \mathbb{T}_{c2}} (x_{t_1,t_2,s} + x_{t_2,t_1,s}) - max_c \quad t_1 \neq t_2, \forall s \in \mathbb{S}_c \quad (22)$$

GA1 Constraints Each GA1 triggers a deviation d_c calculated as the sum of matches of the set $\mathbb{M}_c$ in field "meetings" which occur in set of slots S_c in field "slots" under min_c in field "min" or over max_c in field "max".

$$d_c = \sum_{s \in \mathbb{S}_c} \sum_{t_1, t_2 \in \mathbb{T}_c} x_{t_1,t_2,s} - max_c \tag{23}$$

$$d_c = \sum_{s \in \mathbb{S}_c} \sum_{t_1, t_2 \in \mathbb{T}_c} x_{t_1,t_2,s} + min_c \tag{24}$$

BR1 Constraints Each BR1 with t_c the team in "teams" field, triggers a deviation d_c equal to the sum of teams t_c breaks in set of slots $\mathbb{S}_c$ in "slots" field over max_c in "max" field.

$$d_c = \sum_{s \in \mathbb{S}_c} z_{t_c,s} - max_c \tag{25}$$

BR2 Constraints In all instances where a BR2 constraint exists field "teams" contains all teams and field "slots" contains all slots except the first slot (as a team cannot have a break in the first slot). As such, a BR2 constraint triggers a deviation d_c equal to the sum of all breaks of all teams over max in "max" field.

$$d_c = \sum_{t \in \mathbb{T}} \sum_{s \in \mathbb{S}} z_{t,s} - max_c \tag{26}$$

FA2 Constraints In all instances where an FA2 constraint exists field "teams" contains all teams and field "slots" contains all slots. As such, an FA2 constraint triggers deviations d_c for each pair of teams equal to the largest difference in played home games over all slots more than $intp$ in "intp" field.

$$d_c = \begin{matrix} max \\ s \in \mathbb{S} \end{matrix} \left(\sum_{m=1}^{s} y_{i,m} - \sum_{m=1}^{s} y_{j,m} - intp; 0 \right) \quad \forall i, j \in \mathbb{T}, i < j \tag{27}$$

SE1 Constraints For SE1 we need to keep track of the distance between matches and rematches for all combinations of the set of teams $\mathbb{T}_c$ in field "teams". Each combination of teams triggers a deviation d_c equal to the sum of the number of time slots less than min in "min" field between the match and the rematch.

$$d_c = \left| \sum_{s \in \mathbb{S}} s * x_{t_1,t_2,s} - \sum_{s \in \mathbb{S}} s * x_{t_2,t_1,s} \right| - min_c \quad t1 \neq t2, \forall t_1, t_2 \in \mathbb{T}_c \tag{28}$$

Objective Function Hard constraints must not generate any deviation. Soft constraints' deviations are multiplied by p_c denoted by the field "penalty" and summed. Deviations under zero are ignored.

$$min \quad \sum_{c \in \mathbb{C}} d_c * p_c \tag{29}$$

Employing the CP/SAT Solver of ORTools The overly constrained nature of sports scheduling made it difficult for traditional Constraint Programming solvers to even reach a feasible solution let alone a good one. In our approach we used the ORTools [15] CP/SAT solver that allowed us to formulate the problem using CP terms. The distinctiveness of the CP/SAT solver is that it reformulates internally the CP model into a SAT (satisfiability) model that seems to be better adapted to the nature of the sports scheduling problem.

4.4 A Hybrid Approach

An initial solution satisfying the base constraints is constructed using CP/SAT and Hard constraints are perceived as soft and Soft constraints are ignored. The Simulated Annealing process tries to bring the solution in the feasible space. If a feasible solution is found then Hard constraints become mandatory and Soft constraints are activated. Each time the simulated annealing process terminates an improvement process using CP/SAT attempts to improve the current solution. In order to achieve this we randomly select a number of teams, or a number of slots, or a number of games, or some combination of the above and keep them fixed while the rest of the current solution is allowed to change. In Figure 1 a flowchart of the process is presented.

5 Experimental Results

5.1 Datasets

The problem instances of ITC2021[4] are formatted with the RobinX XML data format [20]. The instances were released in three phases (Early, Middle, Late) and each set contains 15 instances. All instances are in double round robin format of 16, 18 or 20 teams. Some instances contain the Phase rule. Not all constraints make an appearance in every tournament.

5.2 Results

The hybrid process was able to produce solutions for 37 out of 45 instances. The objective of the solutions can be seen in Table 7. Solution files are available at our github[5] repository.

6 Conclusions

Sports scheduling has several facets that make it an interesting and difficult problem. Sports scheduling problems are proved to be, in practice (and in theory), hard to solve. Sometimes even finding a feasible solution or proving that

[4] https://www.sportscheduling.ugent.be
[5] https://bit.ly/3wtrW4i

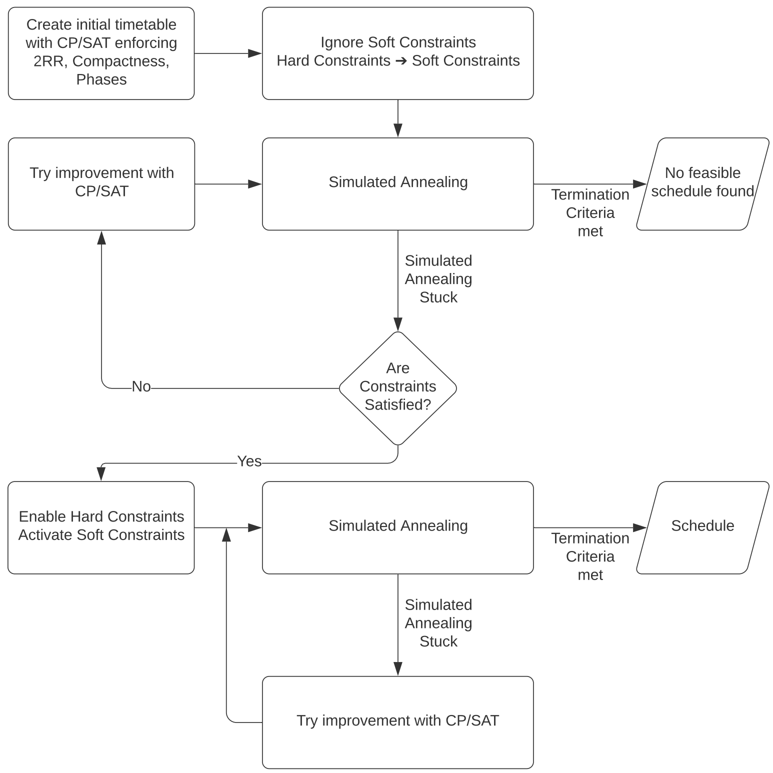

Fig. 1. Flowchart of the hybrid process.

such a solution does not exist is extremely challenging. We had the opportunity to assert this during our participation in the ITC2021 competition.

In this paper, an approach to solving the problem was presented that involved modeling of the problem using Constraint Programming. A Simulated Annealing solver employing small and large moves was implemented. Small moves, are often inspired by the perspective as a graph of the problem, make schedule changes that keep the schedule feasible, but are rather local. Large moves fix randomly selected teams, matches or slots and let the other ones free to move. We managed to receive good results and we firmly believe that our approach can be even more successful in addressing sports scheduling problems by using more processing power and more sophisticated strategies than the random selection for fixed objects.

Acknowledgements We acknowledge support of this work by the project "Dioni: Computing Infrastructure for Big-Data Processing and Analysis." (MIS

Table 7. Results after three hours of execution time for each instance using the hybrid process. Objective is presented as the tuple (deviation of hard constraints, penalty of soft constraints).

Instance	Objective	Instance	Objective	Instance	Objective
Early 1	0, 512	Middle 1	17, -	Late 1	0, 2234
Early 2	0, 266	Middle 2	48, -	Late 2	0, 5680
Early 3	0, 1354	Middle 3	0, 12170	Late 3	0, 3004
Early 4	6, -	Middle 4	0, 7	Late 4	0, 0
Early 5	5, -	Middle 5	0, 732	Late 5	39, -
Early 6	0, 3957	Middle 6	0, 1900	Late 6	0, 1440
Early 7	0, 9644	Middle 7	0, 2792	Late 7	0, 3009
Early 8	0, 1614	Middle 8	0, 301	Late 8	0, 1375
Early 9	0, 448	Middle 9	0, 1015	Late 9	0, 1108
Early 10	32, -	Middle 10	1, -	Late 10	6, -
Early 11	0, 8189	Middle 11	0, 2956	Late 11	0, 511
Early 12	0, 1025	Middle 12	0, 1596	Late 12	0, 7218
Early 13	0, 380	Middle 13	0, 780	Late 13	0, 3576
Early 14	0, 63	Middle 14	0, 1619	Late 14	0, 1650
Early 15	0, 4470	Middle 15	0, 1833	Late 15	0, 80

No. 5047222) which is implemented under the Action “Reinforcement of the Research and Innovation Infrastructure”, funded by the Operational Programme “Competitiveness, Entrepreneurship and Innovation” (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

References

1. Alarcón, F., Durán, G., Guajardo, M., Miranda, J., Muñoz, H., Ramírez, L., Ramírez, M., Sauré, D., Siebert, M., Souyris, S., Andrés, W., Rodrigo, W.Y., Gonzalo, Z.: Operations research transforms the scheduling of Chilean soccer leagues and South American world cup qualifiers. Interfaces **47**(1), 52–69 (2017)
2. Briskorn, D.: Feasibility of home–away-pattern sets for round robin tournaments. Operations Research Letters **36**(3), 283–284 (2008)
3. Chen, J., Dong, D.: Research on the general method of round robin scheduling. In: Advances in Multimedia, Software Engineering and Computing Vol. 2, pp. 393–399. Springer (2011)
4. Costa, F.N., Urrutia, S., Ribeiro, C.C.: An ILS heuristic for the traveling tournament problem with predefined venues. Annals of Operations Research **194**(1), 137–150 (2012)
5. Delahaye, D., Chaimatanan, S., Mongeau, M.: Simulated annealing: From basics to applications. In: Handbook of metaheuristics, pp. 1–35. Springer (2019)
6. Durán, G., Guajardo, M., Sauré, D.: Scheduling the South American Qualifiers to the 2018 FIFA World Cup by Integer Programming. European Journal of Operational Research **262**(3), 1109–1115 (2017)

7. Goossens, D., Spieksma, F.: Scheduling the Belgian soccer league. Interfaces **39**(2), 109–118 (2009)
8. Gunnarsdóttir, E.L.: An integer programming formulation for scheduling of the Icelandic football league. Ph.D. thesis, Reykjavík University (2019)
9. Januario, T., Urrutia, S.: A new neighborhood structure for round robin scheduling problems. Computers & Operations Research **70**, 127–139 (2016)
10. Januario, T., Urrutia, S., Ribeiro, C.C., De Werra, D.: Edge coloring: A natural model for sports scheduling. European Journal of Operational Research **254**(1), 1–8 (2016)
11. Lambrechts, E., Ficker, A., Goossens, D.R., Spieksma, F.C.: Round-robin tournaments generated by the circle method have maximum carry-over. Mathematical Programming **172**(1), 277–302 (2018)
12. Lewis, R., Thompson, J.: On the application of graph colouring techniques in round-robin sports scheduling. Computers & Operations Research **38**(1), 190–204 (2011)
13. Miyashiro, R., Matsui, T.: Round-robin tournaments with a small number of breaks. Department of Mathematical Informatics, The University of Tokyo, Mathematical Engineering Technical Reports METR **29**, 2003 (2003)
14. Nurmi, K., Goossens, D., Kyngäs, J.: Scheduling a triple round robin tournament with minitournaments for the Finnish national youth ice hockey league. Journal of the Operational Research Society **65**(11), 1770–1779 (2014)
15. Perron, L., Furnon, V.: OR-tools, https://developers.google.com/optimization/
16. Ribeiro, C.C.: Sports scheduling: Problems and applications. International Transactions in Operational Research **19**(1-2), 201–226 (2012)
17. Ribeiro, C.C., Urrutia, S.: Scheduling the Brazilian soccer tournament: Solution approach and practice. Interfaces **42**(3), 260–272 (2012)
18. Trick, M.A.: A schedule-then-break approach to sports timetabling. In: International conference on the practice and theory of automated timetabling. pp. 242–253. Springer (2000)
19. Van Bulck, D., Goossens, D., Belien, J., Davari, M.: The fifth international timetabling competition (ITC 2021): Sports timetabling. In: MathSport International 2021. pp. 117–122. University of Reading (2021)
20. Van Bulck, D., Goossens, D., Schönberger, J., Guajardo, M.: RobinX: A three-field classification and unified data format for round-robin sports timetabling. European Journal of Operational Research **280**(2), 568–580 (2020)
21. Van Laarhoven, P., Aarts, E.: Simulated annealing: theory and applications. Dordrecht. Reidel Pub. Comp., Netherlands (1987)