A Fix-and-Optimize Heuristic for the ITC2021 Sports Timetabling Problem

George H.G. Fonseca · Túlio A.M. Toffolo

1 Introduction

This extended abstract briefly describes a fix-and-optimize heuristic for the Fifth International Timetabling Competition (ITC2021), which considered a challenging and realistic Sports Timetabling Problem. The ITC2021 problem consists basically of the assignment of games to rounds in a double round-robin tournament considering many constraints. An even number N of teams is considered, meaning there is a total of $2N-2$ rounds with every team playing exactly once at each round. In total, the ITC2021 problem imposes up to nine different constraints which represent common situations in the real-world. Two types of constraints are considered: hard constraints (H), which must be satisfied at all times, and soft constraints (S), whose violation is penalized in the objective function. The nine different constraints were categorized into five groups by ITC2021 organizers and described by Van Bulck et al. (2021) as:

1. **Capacity constraints** (CA): force a team to play home or away and regulate the total number of games played by a team or group of teams.
2. **Game constraints** (GA): enforce or forbid specific assignments of a game to rounds.
3. **Fairness constraints** (FA): prevent an unbalanced timetable concerning home games, travel distances, etc.

George H. G. Fonseca
Department of Computing and Systems – DECSI
Federal University of Ouro Preto, João Monlevade/MG, Brazil
E-mail: george@ufop.edu.br

Túlio A. M. Toffolo
Department of Computing
Federal University of Ouro Preto, Ouro Preto/MG, Brazil
E-mail: tulio@toffolo.com.br

4. **Break constraints** (BR): regulate the frequency and timing of breaks in a competition; we say that a team has a break if it has two consecutive home games, or two consecutive away games.
5. **Separation constraints** (SE): regulate the number of rounds between consecutive games involving the same teams.

As with most international challenges, a diverse set of benchmark instances was proposed. This is a particularly relevant contribution in the field since most papers addressing similar problems report case studies, resulting in limited comparison among different authors. The availability of benchmark instances may reduce this issue. For further details concerning the ITC2021 problem and the proposed instances, we refer the reader to Van Bulck et al. (2021).

2 Proposed algorithm

We initially formulated the ITC2021 problem as an integer program with three-indexed decision variables $x_{i,j,k}$, which take value 1 if game (i, j) is assigned to round k and 0 otherwise. However, as expected, most of the benchmark instances resulted in models which commercial solvers were not capable of solving within the runtime limit of 10 hours. While this result clearly motivated us to employ heuristics, the proposed integer programming formulation remained as one of the main components of our proposed algorithm, which may be categorized as a matheuristic.

Matheuristics are heuristics that take advantage of the power of mathematical programming (MP) solvers to tackle hard combinatorial optimization problems. More specifically, fix-and-optimize algorithms are matheuristics that iteratively employ a mathematical programming solver to optimize a small sub-problem while the remainder of the problem is fixed.

Algorithm 1 presents the proposed approach. Note that this algorithm is executed twice: first to obtain a feasible solution and then to optimize (improve) this solution. In the first execution, an initial solution is given by the the Polygon Method (Ribeiro and Urrutia, 2007). For the second execution, the feasible solution obtained in the first execution is given as input. Lines 1 and 2 load the MP model, initial solution and decision variables. Note that hard constraints are modelled as soft constraints in the first execution. Stopping criteria consist of a time limit or proven optimality (line 3). The solution is optimal if it has zero cost or if its sub-problem size matches the size of the problem and solver status is optimal. Lines 4 to 17 select the variables to be optimized at each iteration. We considered two ways of releasing variables: neighborhood $\mathcal{N}^R$, which randomly selects n rounds to be optimized (lines 6 to 11), and neighborhood $\mathcal{N}^T$, which randomly selects n teams to be optimized (lines 13 to 17). To allow venue exchange, in neighborhood $\mathcal{N}^R$ we also release the variable related to the inverse venue game of games that occur in one of the selected rounds (line 10). Line 18 fixes the non-selected variables to their current value. Line 19 solves the MP model while line 20 releases the variables for the next iteration. Finally, lines 21 to 24 adjust the sub-problem size.

Algorithm 1: GOAL Solver

```
Input: (i) Problem instance ℙ; (ii) Initial solution s_0; (iii) Sub-problem size n;
       (iv) Time limit t_max; (v) Iteration time limit t_it
Output: (i) Best solution s found.
𝕄 ← Load mathematical model for ℙ
𝒳 ← Load variables of 𝕄 with solution s_0
while elapsed time ≤ t_max and optimal solution has not been found do
    𝒱 ← ∅
    if Random() ≤ 0.5 then
        rounds ← 0
        while rounds < n do
            r ← Randomly select a non-selected round from ℙ
            𝒱 ← 𝒱 ∪ Variables related to round r
            𝒱 ← 𝒱 ∪ Variable related to the inversed venue game of the ones that
              occur in round r
            rounds ← rounds + 1
    else
        teams ← 0
        while teams < ⌊n/2⌋ do
            t ← Randomly select a non-selected team from ℙ
            𝒱 ← 𝒱 ∪ Variables related to team t
            teams ← teams + 1
    Fix variables 𝒳 \ 𝒱 to their current value
    (s, status) ← Solve 𝕄 with time limit t_it
    Release fixed variables in 𝕄
    if status = Optimal then
        n ← n + 1
    else
        n ← n − 1
return s
```

3 Preliminary experiments

The proposed approach was implemented in Java 16. Gurobi 9.1 (Gurobi Optimization, LLC, 2021) was employed to solve sub-problem formulations. The computational experiments were executed on an Intel® Xeon E5620 2.40GHz with 120GB RAM running CentOS Linux 7. Each sub-problem runtime limit was set to 100 seconds, while initial sub-problem size n was set to 10, meaning 10 rounds for $\mathcal{N}^R$ and 5 teams for $\mathcal{N}^T$. We run the experiments with a 24-hour time limit. Table 1 presents the best solutions found by our solver along with the best known solutions (BKS) among all submitted by the 13 teams that participated in ITC2021[1]. For instances in which we could not find feasible solutions, hard (H) and soft (S) costs are displayed as H_S. Solutions marked with a ⊛ are proven optimal.

[1] Reported at `https://www.sportscheduling.ugent.be/ITC2021/instances.php`

Table 1 Best solutions found by the proposed approach for ITC2021.

Instance	Our Best	BKS	Instance	Our Best	BKS	Instance	Our Best	BKS
Early1	421	362	Middle1	5_6039	5177	Late1	2073	1969
Early2	309	160	Middle2	13_7232	7381	Late2	4_6346	5400
Early3	1146	1012	Middle3	9837	9701	Late3	2474	2369
Early4	2_1738	512	Middle4	⊛ **7**	7	Late4	⊛ **0**	0
Early5	13_4631	3127	Middle5	543	413	Late5	12_2402	1939
Early6	4088	3352	Middle6	1630	1125	Late6	1082	923
Early7	6434	4763	Middle7	2394	1784	Late7	2333	1558
Early8	**1064**	1064	Middle8	200	129	Late8	1165	934
Early9	538	108	Middle9	1050	450	Late9	1219	563
Early10	7_4963	3400	Middle10	1537	1250	Late10	13_3559	1988
Early11	5127	4436	Middle11	2798	2511	Late11	361	207
Early12	890	380	Middle12	1007	911	Late12	4786	3689
Early13	331	121	Middle13	430	253	Late13	**1820**	1820
Early14	84	4	Middle14	1682	1172	Late14	1562	1206
Early15	4196	3368	Middle15	1089	495	Late15	160	20

4 Conclusions

We briefly presented a two-neighborhood fix-and-optimize approach for the ITC2021 Sports Timetabling Problem. Limited attention has been given to fix-and-optimize methods for Sports Scheduling in the literature, despite the strong results obtained by such methodology when considering other scheduling problems. Preliminary experiments resulted in feasible solutions for 37 out of 45 instances. We found the best overall solution for 4 instances and proved optimality for 2 of them. These are encouraging results given the difficulty of the problem: even finding feasible solutions is already a challenge for some instances.

We believe there is still room for improvement in the proposed approach. Smarter ways of selecting sub-problems may be proposed; integration with usual heuristic neighborhoods can be explored; and parameter tuning may improve the algorithm's overall performance. Moreover, this approach can heavily benefit from improved formulations. First-break-then-schedule or first-schedule-then-break decompositions may be incorporated as well.

References

Gurobi Optimization, LLC (2021). Gurobi Optimizer Reference Manual.

Ribeiro, C. C. and S. Urrutia (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research 179*(3), 775–787.

Van Bulck, D., D. Goossens, J. Beliën, and M. Davari (2021). The fifth international timetabling competition (ITC 2021): Sports timetabling. In *Proceedings of MathSport International 2021 Conference, MathSport, in press.*, pp. 117–122.