Multi-Neighborhood Simulated Annealing for the Sport Timetabling Competition ITC2021

Roberto Maria Rosati · Matteo Petris ·
Luca Di Gaspero · Andrea Schaerf

1 Introduction

Sport timetabling is an active research field, mainly due to the commercial interest in the maximization of fan attendance (in person or remotely) to sport events. Among the various possible structures for sport competitions, the round-robin tournament is the most frequently used for most team sports.

We describe in this paper the solver that we developed for the Sport Timetabling Competition ITC2021, a three-stage Simulated Annealing approach, that makes use of a portfolio of six different neighborhoods. Five of them are classical ones, already proposed in the literature, whereas the sixth one, named *PartialSwapTeamsPhased*, is a variant of one of them that we specifically designed to deal with phased instances. Our solver has many parameters and it has been tuned using the F-RACE procedure (Birattari et al., 2010), upon a set of experimental configurations designed using the Hammersley point set (Hammersley and Handscomb, 1964).

Overall, the final outcome is that the three-stage Simulated Annealing solver is able to find a feasible solution on 44 out of 45 instances and ranked second in both the first competition milestone and the final round.

2 Related Work

Interest in Sport Timetabling started growing from the 70s, with initial research by Gelling (1973), Russell (1980), Wallis (1983), de Werra (1981), and de Werra et al. (1990). Due to its complexity, (Rosa and Wallis, 1982; Dinitz et al., 1994), several metaheuristic and heuristic algorithms have been proposed for the Sport Timetabling Problem throughout the years. During the years 2000s, new neighborhoods for local-search-based metaheuristics were developed by Ribeiro and Urrutia (2004), Anagnostopoulos et al. (2006) and Di Gaspero and Schaerf (2007), as a consequence of rising interest in the Traveling Tournament Problem (Easton et al., 2001). More recent contributions to (meta)heuristic methods for Sport Timetabling are found in Lewis and Thompson (2011), Costa et al. (2012) and Januario and Urrutia (2016). Finally, Van Bulck et al. (2020b) proposed a unified data format for the round-robin sports timetabling, named RobinX, also employed in the Sport Timetabling Competition ITC2021 (Van Bulck et al., 2021). For a more complete bibliographic revision for sport timetabling we redirect the reader to Rasmussen and Trick (2008) and Kendall et al. (2010).

Roberto Maria Rosati, Luca Di Gaspero, and Andrea Schaerf
DPIA, University of Udine, via delle Scienze 206, 33100 Udine, Italy
E-mail: {robertomaria.rosati,luca.digaspero,andrea.schaerf}@uniud.it

Matteo Petris
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
E-mail: matteo.petris@inria.fr

3 Problem Description

Many variants of the round-robin tournament problem have been discussed in the literature. We consider here the version proposed for the International Timetabling Competition ITC2021 (Van Bulck et al., 2021), which takes into account five types of constraints collected from real-world cases: capacity constraints, game constraints, break constraints, fairness constraints and separation constraints. This formulation has the peculiarity that every single specific constraint can be stated as either hard or soft, as they may express fundamental properties of the timetable and must be satisfied (*hard* version), or they may express preferences and can be violated (*soft* version). Another characteristic of the ITC2021 formulation is that it has abandoned the classical *mirrored* structure in which the second leg is identical to the first one, with home and away positions swapped. That is, the structure of ITC2021 instances is either completely free or *phased*, meaning that a team has to match all other teams in each leg (but not in the same order).

4 Solution Method

We designed a three-stage multi-neighborhood Simulated Annealing for the solution of the problem. As search space we consider the set of all two-leg round-robin timetables. The multi-neighborhood is a hexamodal neighborhood made up by a portfolio of six different local search neighborhoods, which are specifically tailored for the sport timetabling problem. Five of them, called *SwapHomes*, *SwapTeams*, *SwapRounds*, *PartialSwapTeams*, and *PartialSwapRounds*, are adaptations of classical ones from Ribeiro and Urrutia (2004), Anagnostopoulos et al. (2006), and Di Gaspero and Schaerf (2007). The sixth one is a novel neighborhood called *PartialSwapTeamsPhased*, specifically designed to deal with phased instances. It is based on the concept of *mixed phase*, which is a partition of the timetable in two subsets, named mixed legs, where each couple of teams play together, respectively, for the first and for the second time. This definition is independent from the current satisfaction of the phase constraint. The move considers two teams and a set of rounds and swaps the positions of the two teams in the matches in the given set of rounds. A prerequisite is that the matches involved in the move must all belong to the same mixed leg. In this way, the move *PartialSwapTeamsPhased* swaps a subset of teams inside one of the two mixed legs and it is invariant with respect to the phase.

The metaheuristic employed is basically the classical Simulated Annealing defined by Kirkpatrick et al. (1983). The search is executed is three distinct sequential stages. Specifically, the first stage starts its search either from a random or from a greedy solution, the second and the third stages are warm-started with the output of the previous stage. The differences between the stages consist in the restrictions applied to the search space and in the exclusion or inclusion of certain constraints.

5 Experimental Results

Our code was developed in C++ and compiled with GNU g++ version 9.3.0 on Ubuntu 20.04.2 LTS. The tuning phase was partially performed on a cluster of virtual machines provided by the CINECA consortium. All the other experiments presented in this section were run on a machine equipped with AMD Ryzen Threadripper PRO 3975WX processor with 32 cores, hyper-threaded to 64 virtual cores, with base clock frequency of 3.5 GHz, and 64 GB of RAM. In both settings, one single virtual core is used for each experiment.

Table 1 reports the results obtained by the solver. The column *Best solution found* reports the best solution that our solver was able to find in all experiments. Some of these values are those that we submitted to the ITC2021 competition, others have been found in later experiments. When no feasible solution has been found, the number of hard violations followed by a letter H is reported. Next columns, labeled *Average values*, report the data obtained in a set of experiments that we run independently from the competition, in order to extract information on the average behavior of the algorithm in its final configuration. At least 48 runs per instance were performed to collect these data. Columns *Cost* and *Time* report, respectively, the average values of the objective function and the average time needed for a complete run of the three stages. Regarding the average cost, the value is computed only on feasible solutions. Column *Feasible* reports the ratio between feasible solutions and total runs. Finally, column *Best known cost* contains the best known results at the moment this

Instance	Best solution found	Average values			Best known cost
		Cost	Time (s)	Feasible	
Early_1	423	540.7	5667	1.00	362
Early_2	318	384.6	14843	1.00	145
Early_3	1068	1176.5	12194	1.00	992
Early_4	556	1007.8	8759	0.56	507
Early_5	4117	-	28517	0.00	3127
Early_6	3927	4543.0	35161	1.00	3325
Early_7	5205	6721.7	37486	1.00	4763
Early_8	**1051**	1151.9	21394	1.00	1051
Early_9	132	228.7	10324	1.00	108
Early_10	4986	-	35856	0.00	3400
Early_11	4526	5784.5	43692	1.00	4426
Early_12	1010	1200.2	14726	1.00	380
Early_13	173	233.8	19675	1.00	121
Early_14	63	82.3	5616	1.00	4
Early_15	3556	3945.8	46714	1.00	3362
Middle_1	5657	6075.0	26290	0.06	5177
Middle_2	5H	-	26890	0.00	7381
Middle_3	**9542**	11403.1	44748	0.23	9542
Middle_4	16	33.0	5660	1.00	7
Middle_5	510	624.4	6223	1.00	413
Middle_6	1701	2186.3	21350	1.00	1120
Middle_7	2203	2452.7	16303	1.00	1783
Middle_8	136	196.6	19717	1.00	129
Middle_9	640	772.1	17610	1.00	450
Middle_10	1357	1687.5	14432	1.00	1250
Middle_11	2696	2996.5	43876	1.00	2446
Middle_12	950	1054.2	14599	1.00	911
Middle_13	362	479.3	15687	1.00	252
Middle_14	**1172**	1304.6	37483	1.00	1172
Middle_15	985	1099.7	8704	1.00	485
Late_1	2021	2372.7	20242	1.00	1922
Late_2	5715	6085.5	41432	0.49	5400
Late_3	2457	2718.0	18327	1.00	2369
Late_4	**0**	0.0	2354	1.00	0
Late_5	2341	-	9190	0.00	1923
Late_6	930	1121.3	7121	1.00	923
Late_7	1765	2226.5	22959	1.00	1558
Late_8	997	1155.3	11285	1.00	934
Late_9	715	881.2	25963	1.00	563
Late_10	2571	3527.3	32511	0.05	1945
Late_11	207	289.3	15891	1.00	202
Late_12	3944	4830.6	35513	1.00	3428
Late_13	1868	2285.5	21006	1.00	1820
Late_14	**1202**	1326.3	39160	1.00	1202
Late_15	60	82.8	6434	1.00	20

Table 1: Best and average results

article is written, according to data published on the website of the competition (Van Bulck et al., 2020a). When the current known best was determined by our solver, the value in the corresponding column is marked in bold. Overall, our solver could find at least one feasible solution on 44 out of 45 instances. According to data, in its final configuration it manages to determine very easily a feasible solution on 36 instances, which are characterized by a feasibility ratio of 1.00, as it can be observed in column *Feasible* of Table 1. The other instances appear to be harder to solve for the algorithm. In particular, instances Early_5, Early_10, Middle_2, and Late_5 result to be considerably challenging, as feasible solutions are found just sporadically.

We also assessed the impact of the new neighborhood *PartialSwapTeamsPhased*. To do so, we run an additional set of experiments on phased instances with and without making use of *PartialSwapTeamsPhased*. We highlight that employing the new neighborhood *PartialSwapTeamsPhased* brings benefit to the majority of the 22 phased instances: in 17 of these the average cost improvement is 4.24%. One of the remaining five was solved to feasibility only in the configuration that employs *PartialSwapTeamsPhased*. The other four are not solved by any of the two configurations in the given number of runs.

6 Conclusions

In this study, we considered the version of the Sport Timetabling Problem proposed for the ITC2021 competition. We tackled the problem employing a three-stage multi-neighborhood Simulated Annealing approach, which makes use of six different neighborhoods. In particular, the neighborhood that we named PartialSwapTeamsPhased is a novel contribution. Finally, we performed a parameter tuning for the solver using the F-RACE procedure that allowed us to find a set of parameters values for this problem.

This approach managed to find a feasible solution for 44 out of the 45 instances proposed by the competition. Feasible solutions were found rather easily for most of the instances, however the metaheuristic struggled to produce feasible solutions for certain instances, even in long execution times. The results obtained by the Simulated Annealing approach allowed us to rank second out of 13 participants in the final ranking of the competition.

Future work will be devoted to improve the results and performances on both the considered instances and on other benchmark instances for round-robin tournament. We think that relevant advancements can be achieved through a wider study and application of the PartialSwapTeamsPhased neighborhood on a larger set of instances. Possible research directions may also include the definition and integration of new neighborhoods in the Simulated Annealing algorithm, and the implementation and evaluation of new greedy techniques to generate different initial solutions, not restricted to the canonical pattern. Further research may also be committed to develop a matheuristic approach, such as Large Neighborhood Search (LNS), which embeds exact methods in our Simulated Annealing algorithm.

Acknowledgments.

The authors thank the CINECA consortium for the access to the Cloud Computing resources under the Italian Super Computing Resource Allocation Project grant IA4ITC_C.

The authors are also grateful to the ITC2021 organizers David Van Bulck, Dries Goossens, Jeroen Beliën, and Morteza Davari, and to the anonymous reviewers for their very helpful and detailed comments.

References

Anagnostopoulos A, Michel L, Van Hentenryck P, Vergados Y (2006) A simulated annealing approach to the traveling tournament problem. Journal of Scheduling 9(2):177–193

Birattari M, Yuan Z, Balaprakash P, Stützle T (2010) F-race and iterated F-race: An overview. In: Experimental methods for the analysis of optimization algorithms, Springer, Berlin, pp 311–336

Costa FN, Urrutia S, Ribeiro CC (2012) An ils heuristic for the traveling tournament problem with predefined venues. Annals of Operations Research 194(1):137–150

Di Gaspero L, Schaerf A (2007) A composite-neighborhood tabu search approach to the traveling tournament problem. Journal of Heuristics 13(2):189–207

Dinitz JH, Garnick DK, McKay BD (1994) There are 526,915,620 nonisomorphic one-factorizations of k_{12}. Journal of Combinatorial Design 2:273–285

Easton K, Nemhauser G, Trick M (2001) The traveling tournament problem description and benchmarks. In: Seventh International Conference on the Principles and Practice of Constraint Programming (CP 99), Springer-Verlag, LNCS, vol 2239, pp 580–589

Gelling EN (1973) On 1-factorizations of the complete graph and the relationship to round robin schedules. PhD thesis

Hammersley JM, Handscomb DC (1964) Monte Carlo methods. Chapman and Hall, London

Januario T, Urrutia S (2016) A new neighborhood structure for round robin scheduling problems. Computers & Operations Research 70:127–139

Kendall G, Knust S, Ribeiro C, Urrutia S (2010) Scheduling in sports: An annotated bibliography. Computers and Operations Research 37(1):1–19

Kirkpatrick S, Gelatt D, Vecchi M (1983) Optimization by simulated annealing. Science 220:671–680

Lewis R, Thompson J (2011) On the application of graph colouring techniques in round-robin sports scheduling. Computers & Operations Research 38(1):190–204

Rasmussen RV, Trick MA (2008) Round robin scheduling–a survey. European Journal of Operational Research 188(3):617–636

Ribeiro CC, Urrutia S (2004) Heuristics for the mirrored traveling tournament problem. In: Proc. of the 5th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2004), pp 323–342

Rosa A, Wallis WD (1982) Premature sets of 1-factors or how not to schedule round robin tournaments. Discrete Applied Mathematics 4:291–297

Russell KG (1980) Balancing carry-over effects in round robin tournaments. Biometrika 67(1):127–131

Van Bulck D, Goossens D, Beliën J, Davari M (2020a) Website of the fifth international timetabling competition (itc 2021): Sports timetabling. `https://www.sportscheduling.ugent.be/ITC2021/`, last accessed: 16/11/2021

Van Bulck D, Goossens D, Schönberger J, Guajardo M (2020b) Robinx: A three-field classification and unified data format for round-robin sports timetabling. European Journal of Operational Research 280(2):568–580

Van Bulck D, Goossens D, Beliën J, Davari M (2021) The fifth international timetabling competition (itc 2021): Sports timetabling. In: MathSport International 2021 Conference, pp 117–122

Wallis WD (1983) A tournament problem. Journal of the Australian Mathematical Society Series B 24:289–291

de Werra D (1981) Scheduling in sports. In: Hansen P (ed) Studies on Graphs and Discrete Programming, North Holland, pp 381–395

de Werra D, Jacot-Descombes L, Masson P (1990) A constrained sports scheduling problem. Discrete Applied Mathematics 26:41–49