An adaptive large neighbourhood search matheuristic for the ITC2021 Sports Timetabling Competition

Antony E. Phillips · Michael O'Sullivan · Cameron Walker

1 Introduction

We outline a model and method for solving the International Timetabling Competition on Sports Timetabling (Van Bulck et al., 2021b). Our algorithm generates a starting solution, and improves it using an adaptive large neighbourhood search (ALNS) using several neighbourhood types. Each neighbourhood subproblem is solved using integer programming, classifying the overall approach as a matheuristic. Similar methods have been effective on other classes of constrained scheduling problems (e.g. Pisinger and Røpke, 2007; Lindahl et al., 2018).

During the competition we generated 3 best-known solutions, and another 17 best-known solutions afterwards by improving other contestants solutions.

2 Modelling & Solution Approach

Our approach initially defines a monolithic integer program (IP) model which fully encodes the ITC2021 problem specification. The primary decision variables define the binary choice of assignment of a single game (ordered pair of teams) to a slot in the double round robin.

These variables are sufficient to prescribe a solution, however to enable modelling of the specified hard and soft constraints, we use two sets of auxiliary variables. The first set represents whether a break has occurred for each team, in each slot, for each game mode. The second set represents the magnitude of violation for each soft constraint, which are used to define the objective function.

Antony E. Phillips (aphi.xc@gmail.com)
Phillips Data Science Ltd

Michael O'Sullivan, Cameron Walker
Department of Engineering Science, University of Auckland

Neighbourhood	Description
Slots	Game-slot variables are free within a subset of slots
Teams	Game-slot variables are free if both teams are within a subset of teams
Teams+	Game-slot variables are free if either team is within a subset of teams
Order	Game-slot variables are free if this game or the reversed game is assigned to this slot in the current solution

Table 1 Neighbourhood types

This simple formulation is unlike most mathematical programs in the sports timetabling literature, which tend to use a multi-stage approach (Rasmussen and Trick, 2008). For example, an initial stage to select allowable home-away patterns, and a latter stage to build the full schedule.

For the instances in ITC2021, the monolithic IP we define can easily be constructed in memory, with the largest instance (Early 15) represented with 28,171 variables, 23,842 constraints and 11.1 million nonzeros. However, the problem structure makes this model intractable to solve to optimality.

2.1 Starting solution

To find a starting solution, we first attempt to solve the monolithic IP (terminating after 8 hours), which may find a feasible solution. If not, we construct a starting solution using a "canonical factorization" from de Werra (1981), which minimises the number of breaks. This solution is guaranteed to satisfy the challenging "BR2" constraint (maximum total number of breaks), but is likely to violate other hard constraints and thus not be feasible. Using the number of hard constraint violations as an objective function to be minimised, we then employ a hill climbing heuristic. Specifically we try all pairs of swaps between teams and slots (e.g. every assigned game for two teams are swapped).

2.2 Improvement Phase

From a starting solution, we iteratively apply an adaptive large neighbourhood search (ALNS), where part of the solution is allowed to be modified while the rest remains fixed. An IP is solved within this neighbourhood subproblem which aims to minimise the number of hard or soft constraint violations, depending whether the current solution is infeasible or feasible respectively. The types of neighbourhood are shown in Table 1.

The selection of which neighbourhood type to use is treated as a multi-armed bandit problem, and addressed using the Upper Confidence Bound (UCB) method. Based on the results from all previous iterations on this instance, the next neighbourhood type ("arm") is chosen as that with the greatest optimistic upper bound on its expected probability of improving the solution ("reward"); see formula (2.10) from Sutton and Barto (2018). This balances between exploring options and exploiting those which have performed well in previous iterations.

The size of the neighbourhood is also adaptive, based on the outcome in the last iteration for this neighbourhood type. Except for the fixed-size "Order" neighbourhood, the size is increased or decreased by 1 unit (i.e. a slot or team) if the last iteration was solved within 5 minutes or if it terminated on a 30 minute time limit respectively. Searching a larger number of small neighbourhoods was found to be more effective than the opposite.

Within each iteration, the specific subset of slots or teams chosen for the neighbourhood is determined based on a randomly selected unsatisfied constraint (hard or soft). This allows the neighbourhood to focus on assignments which contribute to the total penalty of the solution. If the constraint is defined over more slots or teams than required for the neighbourhood, a random subset are chosen. In the converse situation, additional slots or teams are selected at random.

2.3 Results

To solve instances in parallel, we used Google Cloud Platform both to run our algorithm (Compute Engine virtual machines), and to maintain an online database of solutions and attempts (BigQuery).

Over several days, we used 4 "c2-standard-30" virtual machine instances, for a total of 10,686 vCPU hours (Google Cloud, 2021). Therefore, each of the 45 competition instances received an average of 237 vCPU hours, terminating on the total time elapsed. This is approximately similar to 1-2 days of execution on a standard consumer CPU (with 4 to 8 physical cores). All IPs were solved using Gurobi 9.1.1, with the "MIPFocus" parameter set to 1.

Our full set of results are shown in Table 2. The objective values of our best solutions during the competition are given in column "Us-ITC", and the best solutions from all teams in column "Best-ITC". We were able to find a feasible solution to 37 out of 45 instances, of which 3 solutions were the best-known from all submissions (as marked with an asterisk). However, most of our solutions have a notably higher objective than the best-known solutions.

After the competition, we tested the ALNS algorithm on the best-known solutions from all teams, each for 4 hours on a consumer CPU (AMD Ryzen 5900HX). In 17 cases we were able to generate a new best-known solution, with objective shown in column "Us-Post" of Table 2. These solutions are available on the competition website (Van Bulck et al., 2021a). Of the 17 highly optimised starting solutions, 12 were provided by Team Saturn, 3 by Team UoS, 1 by Team Udine and 1 by Team GOAL.

Finally, Figure 1 demonstrates the ALNS algorithm on the 'Early 15' instance. Starting from a feasible solution obtained by solving the monolithic model (with objective of 7,504), the ALNS algorithm reduces the objective value to 4,667 over 1100 iterations. Only the 91 successful iterations and the associated neighbourhood type are shown.

Instance	Objective Us-ITC	Best-ITC	Us-Post
Early 1	666	362	
Early 2	379	160	145
Early 3	1171	1012	992
Early 4	–	512	507
Early 5	–	3127	
Early 6	4821	3352	3325
Early 7	7208	4763	
Early 8	1191	1114	1074
Early 9	447	108	
Early 10	–	3400	
Early 11	6713	4436	4426
Early 12	925	380	
Early 13	382	121	
Early 14	106	4	
Early 15	4667	3368	3362
Middle 1	–	5177	
Middle 2	–	7381	
Middle 3	11235	9701	
Middle 4	7*	7	
Middle 5	681	413	
Middle 6	2026	1125	1120
Middle 7	3317	1784	1783
Middle 8	277	129	
Middle 9	1315	450	
Middle 10	2370	1250	
Middle 11	3143	2511	2446
Middle 12	911*	911	
Middle 13	1044	253	252
Middle 14	1704	1172	
Middle 15	1401	495	485
Late 1	2406	1969	1922
Late 2	–	5400	
Late 3	2900	2369	
Late 4	0*	0	
Late 5	–	1939	1923
Late 6	1310	923	
Late 7	2805	1558	
Late 8	1252	934	
Late 9	1343	563	
Late 10	–	1988	1945
Late 11	376	207	202
Late 12	5542	3689	3428
Late 13	3099	1820	
Late 14	1714	1206	
Late 15	80	20	

Table 2 Full Results

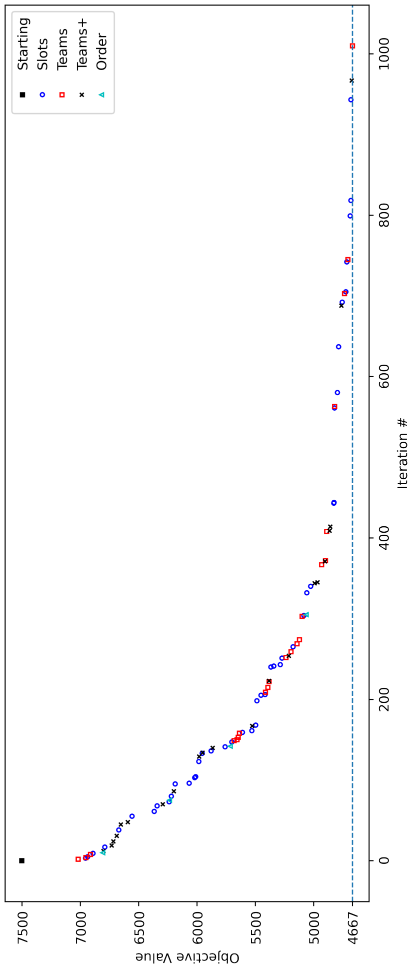

Fig. 1 ALNS Iterations for Early 15

3 Conclusion

Our algorithm performed adequately given its relative simplicity and modest execution time. The method to generate a starting solution was particularly simple, and consequently often could not provide a feasible solution to the ALNS algorithm. Notably, the 8 unsolved instances all include a phased tournament and a hard "BR2" constraint, which add dependencies across the entire solution and are hard to control with the defined ALNS neighbourhood types.

However, when operating on feasible solutions, the ALNS method was able to rapidly improve most solutions an appreciable amount. This led to us finding 3 best-known solutions during the competition, and 17 more after the competition, by improving the solutions from other teams.

The ALNS algorithm could likely be further sped up, as less than 10% of neighbourhoods found an improved solution. This suggests an opportunity for a more targeted choice of neighbourhoods, whether derived analytically or with online learning. The ALNS method could additionally be hybridized with the conventional decomposition approaches in sports timetabling, which add structured home-away patterns and multiple starting solutions.

References

de Werra, D. (1981). Scheduling in Sports. In Hansen, P., editor, *Annals of Discrete Mathematics (11)*, volume 59 of *North-Holland Mathematics Studies*, pages 381–395. North-Holland.

Google Cloud (2021). Machine families — Compute Engine Documentation — Google Cloud. https://cloud.google.com/compute/docs/machine-types. Accessed 2021-07-05.

Lindahl, M., Sørensen, M., and Stidsen, T. R. (2018). A Fix-and-Optimize Matheuristic for University Timetabling. *Journal of Heuristics*, 24(4):645–665.

Pisinger, D. and Røpke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.

Rasmussen, R. and Trick, M. (2008). Round robin scheduling – a survey. *European Journal of Operational Research*, 188:617–636.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press.

Van Bulck, D., Goossens, D., Beliën, J., and Davari, M. (2021a). International Timetabling Competition 2021: Sports Timetabling – website. https://www.sportscheduling.ugent.be/ITC2021.

Van Bulck, D., Goossens, D., Beliën, J., and Davari, M. (2021b). The Fifth International Timetabling Competition (ITC 2021): Sports Timetabling. In *Proceedings of MathSport International 2021 Conference*, MathSport, pages 117–122.