Branch-and-cut for a real-life highly constrained soccer tournament scheduling problem

Guillermo Durán[1], Thiago F. Noronha[2], Celso C. Ribeiro[3], Sebastián Souyris[1], and Andrés Weintraub[1]

[1] Department of Industrial Engineering, University of Chile, Republica 701, Chile.
[2] Department of Computer Science, Catholic University of Rio de Janeiro, Rua Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil.
[3] Department of Computer Science, Universidade Federal Fluminense, Rua Passo da Pátria 156, Niterói, RJ 24210-240, Brazil.
{tfn,celso}@inf.puc-rio.br, {gduran, ssouyris, aweintra}@dii.uchile.cl

1 Introduction

There are 20 teams in the Chilean soccer first division. They take part in two yearly tournaments: *opening* and *closing*. Each tournament is organized in two phases: *qualifying* and *playoffs*. The qualifying phase follows the structure of a single round robin tournament. The teams are evenly distributed over four groups with five teams each. The groups are formed according to the performance of each team in the last tournament. The first four teams are distributed on the four groups. The teams from 5th to 8th place are randomly distributed in different groups. This scheme is repeated until all teams are assigned to a group. At the end of the qualifying phase, the teams that end up in the two first positions of each group qualify for the playoffs.

The National Association of Professional Football (ANFP) is in charge of scheduling the games of the opening and closing tournaments. Good schedules are of major importance for the success of the tournaments, making them more balanced, profitable, and attractive. All schedules were randomly prepared until 2004. Weintraub et al [1] addressed the main drawbacks of such schedules and tackled the problem by integer programming. Their model is applied since 2005. However, the computation times are very high and the solutions produced by the model still lack quality.

In this work, we improve the original integer programming formulation. Valid inequalities are derived and appended to the model. A new branch-and-cut strategy is used to speedup convergence. The main constraints and the objective function are described in Section 2. The solution approach and the branching strategy are summarized in Section 3. Preliminary results on a real-life instance are reported in the last section.

2 Problem statement

A HAP stands for a home-away pattern and defines a sequence of home and away games for a given team. *Popular* teams are those with more fans. *Traditional*

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 398–401. ISBN 80-210-3726-1.

teams are the oldest teams. *Strong* teams are those better qualified in the last tournaments. *Tourist* teams are those from Viña del Mar, Valparaiso, and La Serena. A *classic* is a game between two traditional teams. The constraints of the problem are the following:

- Each team plays against every other team exactly once.
- Every team plays exactly once in each round.
- Each team plays at least nine games at home and nine games away.
- A team may never have two consecutive breaks [5].
- A team may play at most three games at home in any five consecutive rounds.
- Some teams have complementary HAPs (whenever one of them plays at home the other plays away, and vice-versa).
- There may be at most four games in Santiago in any round.
- There may be no breaks in rounds 1, 16, and 18.
- Pairs of excluding teams: if a team plays against one of them at home, then it should play away against the other (and vice-versa).
- Classics should not be played before round 7 or after round 16.
- No team can play two consecutive games against popular teams.
- No team can play two consecutive games against strong teams.
- Each traditional team plays exactly one classic at home.
- Tourist teams should play at least once against a traditional team during the summer rounds.
- Traditional teams cannot play twice in the same week in the same tourist region.
- A team from the Central region cannot play in the same week against a team from the South and another from the North.

Since only the teams in the two first positions of each group qualify for the playoffs, games between teams in the same group are more attractive. Therefore, these games should as much as possible be held in the last rounds. The objective function consists in maximizing the number of games between teams in the same group in the last rounds of the tournament. It is given by the sum of the indices of the rounds where the games between teams of the same group take place.

3 Solution approach and branching strategy

The problem is formulated by integer programming and solved by a branch-and-cut algorithm. Two variables were used in the original model [1]: $x_{ijk} = 1$ if team i plays at home against team j in round k, $x_{ijk} = 0$ otherwise; $y_{ik} = 1$ if team i has a break at round $k + 1$ (i.e., team i plays two consecutive home games or two consecutive away games in rounds k and $k+1$), $y_{ik} = 0$ otherwise.

The new formulation follows the same strategy proposed by Trick [3] and is based on the introduction of a new binary variable: $z_{ik} = 1$ if team i plays at home in round k, $z_{ik} = 0$ otherwise. All HAP constraints are rewritten in terms of this new variable. This formulation is more easily solvable. Furthermore, the new variable plays a major role in the branching strategy. Each round is a perfect

matching in the complete graph whose nodes are the participating teams [2, 4]. Cuts associated with the most violated matching constraints in the linear relaxation are progressively added to the enumeration tree.

The branching strategy plays a major role in the success of a branch-and-cut algorithm. Branching on the x_{ijk} variables is not efficient, since most of them are null in integral solutions. Our branching strategy is based on the z_{ik} variables. Branching on the x_{ijk} variables starts only after all the z_{ik} variables are integral. This strategy implicitly decomposes the solution in two phases. In the first phase the HAPs for each team are computed, while in the second the dates of the games are established. Once the variables z_{ik} are fixed, the branch-and-cut algorithm needs just a few branches on variables x_{ijk} to find a feasible solution.

4 Preliminary results

Two algorithms based on the previous formulation have been proposed and evaluated: the `B&C-ANFP` branch-and-cut algorithm and the `B&B-ANFP` branch-and-bound algorithm without cuts. Both of them were implemented using the library Concert Technology 1.2 and version 8.0 of the CPLEX solver. The computational experiments were performed on a 3 GHz Pentium IV machine with 1 Gbyte of RAM memory. We illustrate the results obtained for the 2005 edition of the opening tournament, comparing them with those reported in [1].

Computation times for solving the linear programming relaxation by different algorithms available with the CPLEX 8.0 package are given in Table 1. The problem is very degenerated and requires the use of perturbations, leading to large computation times. The interior point algorithm was the best solution strategy. We also can see in Table 1 that the new formulation considerably reduced the computation time of the interior points algorithm, making possible an efficient implementation of the cutting plane algorithm.

Table 1. Computation times for solving the linear relaxation.

Strategy	time (s)
Primal simplex	27
Dual simplex	21
Interior points (original formulation)	12
Interior points (variables z_{ik})	4

Results obtained with algorithms `B&B-ANFP` and `B&C-ANFP` are given in Table 2, which reports the value of the objective function, the number of nodes in the enumeration tree, and the integrality gap after some elapsed time. In the beginning, algorithm `B&B-ANFP` finds good solutions faster than `B&C-ANFP`. However, the former was not able to find the optimal solution after 4 hours. On the contrary, the cuts used by algorithm `B&C-ANFP` were able to improve the linear

relaxation bound, which was already equal to the optimal value at the root of the enumeration tree. The number of nodes is much smaller for algorithm B&C-ANFP, that found the optimal solution in less than two hours of computation time.

Table 2. Comparison between algorithms B&B-ANFP and B&C-ANFP.

Elapsed	B&B-ANFP			B&C-ANFP		
time	objective	nodes	gap (%)	objective	nodes	gap (%)
10 minutes	617	140	3.6	474	120	25.9
30 minutes	622	600	2.9	615	330	3.9
1 hour	631	1120	1.4	633	570	1.1
2 hours	633	2190	1.1	640	1560	0.0
4 hours	639	4860	0.2	—	—	—

In Table 3, we compare the results obtained by algorithm B&C-ANFP with those obtained by the approach in [1]. We give the value of the objective function and the integrality gap after 30 minutes and after two hours of computation time (on a 2.4 GHz Pentium IV computer for [1]) for both algorithms. Algorithm B&C-ANFP not only found a better solution (615) very quickly (30 minutes), but also found the optimal solution value (640) after the same time that the approach in [1] took to find a solution value 10.0% away from the optimal.

Table 3. Comparison between algorithms B&C-ANFP and Weintraub et al [1].

Algorithm	time	objective	gap (%)
B&C-ANFP	30 minutes	615	3.9
	2 hours	640	0.0
Weintraub et al [1]	2 hours	576	10.0

References

1. G. Durán, M. Guajardo, J. Miranda, D. Sauré, S. Souyris, A. Weintraub, A. Carmash, and F. Chaigneau. Programación matemática aplicada al fixture de la primera divisón del fútbol chileno. *Revista Ingeniería de Sistemas*, 5:29–46, 2005.
2. M.W. Padberg and M.R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operation Research*, 7:67–80, 1982.
3. M.A. Trick. A schedule-and-break approach to sports scheduling. *Lecture Notes in Computer Science*, 2079:242–253, 2001.
4. M.A. Trick. Integer and constraint programming approaches for round robin tournament scheduling. *Lecture Notes in Computer Science*, 2740:63–77, 2003.
5. S. Urrutia and C.C. Ribeiro. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. *Discrete Applied Mathematics*, to appear.