DEPARTMENT OF OPERATIONS RESEARCH
UNIVERSITY OF AARHUS

Working Paper no. 2006 / 1 2006 / 03 / 21

Scheduling a triple round robin tournament for the best Danish soccer league

Rasmus V. Rasmussen

ISSN 1600-8987

Department of Mathematical Sciences
Telephone: +45 8942 1111
E-mail: institut@imf.au.dk

Building 1530, Ny Munkegade
DK-8000 Aarhus C, Denmark
URL: www.imf.au.dk

Scheduling a triple round robin tournament for the best Danish soccer league

Rasmus V. Rasmussen

Department of Operations Research, University of Aarhus, Ny Munkegade, Building 1530, 8000 Aarhus C, Denmark

Abstract In this paper we present a solution method for the highly constrained problem of finding a seasonal schedule for the best Danish soccer league. The league differs from most sports leagues, since it plays a triple round robin tournament which leads to an uneven distribution of home and away games. The solution method presented here uses a logic-based Benders decomposition in which the master problem finds home-away pattern sets while the subproblem finds timetables. Furthermore, column generation techniques are used to enhance the speed of the master problem. The computational results show that the solution method is capable of solving the problem within reasonable time and the Danish Football Association has decided to use it for scheduling the 2006/2007 season.

Keywords: Timetabling; Sports scheduling; Logic-based Benders decomposition.

1 Introduction

Seasonal schedules for sports leagues are subject to a wide range of often conflicting interests coming from teams, television networks, spectators and the association arranging the tournament. This makes the problem of designing solution methods which are capable of handling these constraints - an interesting and challenging task seen from an operations research perspective. Furthermore, effective methods are very attractive, since the turnover coming from TV rights and spectators may be highly dependent on the number of requirements being satisfied.

This has lead to a significant amount of research dealing with practical applications. Bartsch, Drexl and Kröger [2] have scheduled the Austrian and German soccer league, Croce and Oliveri [4] have scheduled the Italian soccer league and Schreuder [16] has scheduled the Dutch soccer league. Furthermore, de Werra, Jacot-Decombes and Masson [5], Easton [6], Henz [9] and Nemhauser and Trick [13] have scheduled basketball tournaments, Russel and Leung [15] have scheduled a baseball tournament, Ferland and Fleurent [7, 8] and Costa [3] have scheduled the National Hockey League and Armstrong and Willis [1] have scheduled the cricket world cup.

This paper presents an algorithm for finding a seasonal schedule for the best Danish soccer league SAS Ligaen. SAS Ligaen differs from leagues considered in previous work since teams meet three times instead of one or two times. This difference results in a number of additional constraints which must be satisfied in order to get a fair tournament. Furthermore, this work is the first applying the techniques of the pattern generating Benders approach (PGBA) presented by Rasmussen and Trick [14] to a practical application.

In the original version of the PGBA, place constraints were considered but not all the constraints which are present in a real tournament can be included as easily. In this work we present an algorithm which uses the strengths of the PGBA but at the same time allows all the constraints which are considered in SAS Ligaen.

The paper is organized as follows. In Section 2 we present the constraints for SAS Ligaen and give a short introduction to sports scheduling terminology. An outline of the solution method is given in Section 3 and Section 4 describes the algorithm in details. Computational results are reported in Section 5 and we finish with concluding remarks in Section 6.

2 Problem formulation

We face the problem of designing a seasonal schedule for the best Danish soccer league SAS Ligaen. The league consists of 12 teams and has a somewhat unusual structure compared to most sports leagues, since the teams meet three times instead of two. This difference leads to a number of unique constraints in addition to the usual constraints applicable to any sports schedule. Before we explain the constraints in details we will give a brief introduction to the sports scheduling terminology used in the rest of the paper.

2.1 Sports scheduling terminology

A *round robin tournament* is a tournament in which all teams meet a fixed number of times and the tournament considered in this paper is a *triple round robin tournament* meaning that all teams meet three times. The tournament is partitioned into time slots and all teams play exactly one game in each slot. Each game takes place at one of the opponents' venues and the team which plays home is said to play a *home game* while the visiting team plays an *away game*. If a team plays two consecutive home games or two consecutive away games, it is said to have a *break* in the last of the two slots.

The sequence of home and away games for a particular team is called a *home-away pattern* (pattern) and a set containing a pattern for each team is called a *pattern set*. A *timetable* is a table which shows the opponent of each team in each slot. Figure 2.1 shows a pattern set and a timetable for a tournament with 6 teams and breaks are underlined in the pattern set. We say that a pattern set is feasible if a timetable exists which can be used together with the pattern set. In this case the combination of the pattern set and the timetable gives a schedule.

Slot	1	2	3	4	5	6	7	8	9	10
p_1	0	1	0	1	0	1	0	1	0	1
p_2	0	1	$\underline{1}$	0	1	$\underline{1}$	0	$\underline{0}$	1	0
p_3	1	0	1	$\underline{1}$	0	$\underline{0}$	1	0	$\underline{0}$	1
p_4	0	1	0	$\underline{0}$	1	$\underline{1}$	0	1	$\underline{1}$	0
p_5	1	0	1	0	1	0	1	0	1	0
p_6	1	0	$\underline{0}$	1	0	$\underline{0}$	1	$\underline{1}$	0	1

(a)

Slot	1	2	3	4	5	6	7	8	9	10
team 1	6	3	5	2	4	6	3	5	2	4
team 2	5	6	4	1	3	5	6	4	1	3
team 3	4	1	6	5	2	4	1	6	5	2
team 4	3	5	2	6	1	3	5	2	6	1
team 5	2	4	1	3	6	2	4	1	3	6
team 6	1	2	3	4	5	1	2	3	4	5

(b)

Figure 2.1: (a) Pattern set, (b) Timetable.

2.2 Constraints for SAS Ligaen

The major challenge when creating the schedule for a sports league is to satisfy the constraints arising from teams, spectators, TV stations, other tournaments, etc. The constraints are often conflicting and call for a solution method which is able to rank schedules with respect to the number of broken constraints instead of methods searching for schedules which satisfy all constraints.

The constraints are partitioned into *hard constraints* which must be satisfied and *soft constraints* which incur a penalty in case they are violated. Below is an outline of the constraints for SAS Ligaen.

The structure of SAS Ligaen gives rise to a triple round robin tournament with 33 slots and 6 games in each slot. Furthermore, the tournament consists of three single round robin tournaments such that the slots from 1 to 11, the slots from 12 to 22 and the slots from 23 to 33 all form a single round robin tournament. In the rest of the paper the single round robin tournament in the slots 1 - 11 will be referred to as Part 1 and the double round robin tournament in slots 12 - 33 will be referred to as Part 2. In Part 2 all teams must meet all other teams once at home and once at the opponent's venue.

Consecutive constraints limit the number of consecutive home games and consecutive away games to be less than or equal to 2. This is a hard constraint.

Separation constraints give a lower limit k on the number of slots between two games with the same opponents. If $k = 0$ it means that *repeater games* are allowed. But when $k = 1$ there must be at least one slot between slots where the two teams meet. This is a hard constraint.

Best half constraints are hard constraints stating that the 6 teams which finished in the best half of the tournament the preceding year get an extra home game in Part 1. This means that these teams play 6 home games in Part 1, while the other teams play 5 home games.

Ending constraints are hard constraints saying that teams cannot have a break at the last slot.

Place constraints are constraints saying that a specific team wants to play home in a certain slot or away in a certain slot. These constraints are hard or soft depending on the reason. If for instance a stadium is unavailable due to reconstruction or a concert it would be a hard constraint but in case the requirement is imposed due to nearby arrangements it would be a soft constraint.

Game constraints state that at least one game between a specific pair of teams

(i_1, i_2) must be played in a certain set of slots. The game constraints are soft constraints and the set of constraints are denoted C^{Ga}. For a game constraint l, the set T_l^{Ga} denotes the pair of teams (i_1, i_2) and S_l^{Ga} denotes the set of slots in which a game between i_1 and i_2 must be played.

Top team constraints make sure that all non-top teams play at least one home game against one of the top teams in Part 1. In SAS Ligaen two teams are categorized as top teams. This is partly due to good results but also due to a large number of fans which means that revenue from spectators increase when a top team is visiting. Since the revenue goes to the home team, all teams want to play top teams home in Part 1. This is only a concern in Part 1 since all teams meet all other teams once home and once away in Part 2. The top team constraints are soft constraints and the set of non-top teams is denoted T^{To}.

Home constraints are used when a team i_1 has played many away games against another team i_2 in Part 1 in the preceding years. In this case a soft constraint can be added to make sure that team i_1 plays home against i_2 in Part 1. The set of home constraints are denoted C^{Ho} and for each $l \in C^{Ho}$ we use T_l^{Ho} to denote the set of teams (i_1, i_2) where i_1 must play home.

Beginning constraints are soft constraints which state that all teams must have a home game and an away game in the first two slots.

Geographic constraints require that at least one team from a certain area plays home in each slot or at least one team plays away in each slot. These constraints are used to avoid slots where some areas experience a large number of games while others are without games. We let C_H^{Ge} and C_A^{Ge} denote the set of constraints where at least one team must play home and the set of constraints where at least one team must play away while $C^{Ge} = C_H^{Ge} \cup C_A^{Ge}$ denotes the total set of geographic constraints. For each $l \in C^{Ge}$ the set T_l^{Ge} denotes the set of teams from the given area.

Break constraints say that the teams must alternate between home and away games. These are soft constraints and they are broken each time a team has a break.

The objective is to minimize the total penalty imposed by the violated constraints. For each of the soft constraints a coefficient represents the penalty which is added to the objective value if the constraint is violated. The coefficients can be seen in Table 2.1, which gives an overview of the constraints. For each constraint it shows whether the constraint is hard, soft, has influence on Part 1, Part 2, the pattern set or the timetable.

3 Methodology

Due to the complexity of the problem we use a logic-based Benders decomposition strategy [10]. The master problem consists of finding a pattern set and allocate teams to patterns while the subproblem finds a timetable if any exists. If a timetable is found, an optimality cut is added to the master problem and, otherwise, a feasibility cut is added.

In traditional Benders decomposition, the subproblem is a linear programming problem which makes it possible to generate a Benders cut from the dual variables. In our case, the subproblem is an IP problem and therefore we have to use logic-based

Table 2.1: Constraints for SAS Ligaen

Constraint	Hard	soft	Part 1	Part 2	Patt. Set	TT.	Coef.	Con. Set
Structure	×		×	×	×	×	—	—
Consecutive	×		×	×	×		—	—
Separation	×		×	×		×	—	—
Best Half	×		×		×	×	—	—
Ending	×			×	×		—	—
Place	×	×	×	×	×		c^{Pl}	C^{Pl}
Game		×	×	×		×	c^{Ga}	C^{Ga}
Top Team		×	×			×	c^{To}	T^{To}
Home		×	×			×	c^{Ho}	C^{Ho}
Beginning		×	×		×		c^{Be}	—
Geographic		×	×	×	×		c^{Ge}	C^{Ge}
Break		×	×	×	×		c^{Br}	—

Benders cuts instead of traditional Benders cuts. Furthermore, in order to limit the number of feasible solutions to the master problem, we use a column generation strategy to solve the master problem.

This leads to a solution method which decomposes the problem into four steps. In Step 1, we generate patterns, in Step 2, we find a pattern set and allocate teams to patterns, in Step 3, we check feasibility of the pattern set found in Step 2 and finally, in Step 4, we find a timetable. The four steps are visited iteratively during the process.

The algorithm uses a set containing all patterns which have been generated. Initially this set is empty but each time the algorithm goes to Step 1, additional patterns are added unless all feasible patterns have been generated already. In that case, the algorithm stops. The first time Step 1 is used, it generates all patterns with 0 breaks and each time the algorithm returns to Step 1, the number of breaks is increased by one. In this way the best patterns, with respect to the number of breaks, are considered first.

When patterns have been generated in Step 1, we solve an IP problem in Step 2 to find a pattern set and allocate the teams to the patterns. This IP problem is referred to as the master problem, since it resembles the master problem from Benders decomposition. In case the master problem is infeasible, we return to Step 1 where additional patterns are generated and otherwise we go to Step 3.

Step 3 is used to detect infeasible pattern sets and generate logic-based Benders cuts which can be added to the master problem. The strength of a logic-based Benders cut depends on the number of infeasible pattern sets it is able to cut off, and in order to find strong cuts, we need to know why the pattern sets are infeasible. If we have an infeasible pattern set but no knowledge of why it is infeasible we can only prevent the master problem from finding the same solution again. On the other hand, if we know that a pattern set is infeasible because it contains two patterns which cannot be in the same pattern set, we can add a cut which prevents the master problem from finding any pattern set containing both of these two patterns.

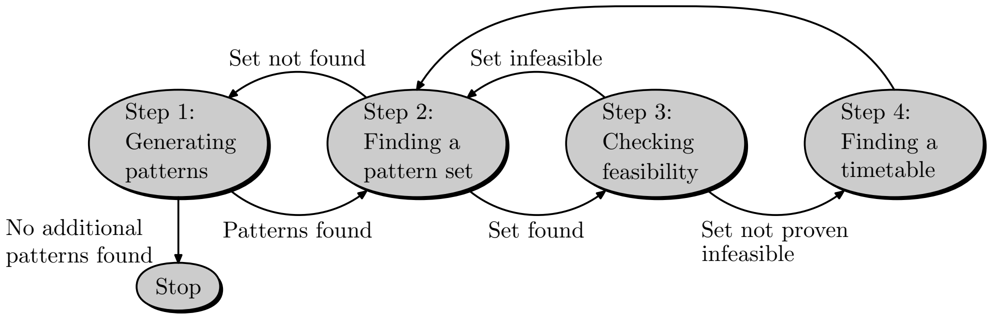

Figure 3.1: Flowchart for the algorithm.

Therefore, Step 3 contains a number of feasibility checks which are used to determine why a pattern set if infeasible. If infeasibility is detected, we return to Step 2 and otherwise, we proceed to Step 4. However, the feasibility checks in Step 3 are not exhaustive meaning that the pattern set might be infeasible anyway.

The problem of finding an optimal timetable for the pattern set found in Step 2 is formulated as an IP model and is referred to as the subproblem. In case the problem is infeasible, it means that the pattern set is infeasible and a logic-based Benders cut is added to the master problem. Otherwise we have found a feasible schedule and an optimality cut can be added to the master problem. In both cases we return to Step 2 after the cut has been added.

The algorithm keeps iterating until the master problem eventually becomes infeasible and all feasible patterns have been generated. When this happens we have either found an optimal solution or proved that the problem is infeasible. Figure 3.1 displays a flowchart showing how the algorithm iterates between the four steps.

4 The algorithm

In this section we give a more detailed description of the four steps discussed in Section 3 but before we do that we need some notation. In the rest of the paper we let n denote the number of teams while T denotes the set of teams. The set $\mathcal{P} = \{1, 2\}$ represents the two parts discussed in Section 2 and S, S^1 and S^2 denote the set of all slots, the set of slots in Part 1 and the set of slots in Part 2, respectively.

4.1 Generating patterns

The partitioning of the tournament into Part 1 and Part 2 is used to reduce the total number of patterns which must be generated. Instead of generating patterns covering all slots, we generate patterns for Part 1 and Part 2 separately.

In addition to reducing the number of patterns, the partitioning also makes the algorithm more flexible. If one part is infeasible, we can generate additional patterns for this part alone without generating extra patterns for the other part. We let P^1 and P^2 denote the sets of patterns which have been generated for Part 1 and Part 2 respectively while B^1 and B^2 denote the current upper bounds on the number of breaks. When additional patterns are generated for Part p, the associated bound B^p

is increased by one and all feasible patterns for Part p with exactly B^p breaks are generated. In order to limit the number of breaks per team, B^1 and B^2 cannot exceed 3 but still this allows up to seven breaks per team.

The generation of patterns has been implemented in OPL Script but the feasible patterns for Part 1 with B^1 breaks correspond to the feasible solutions of the following CP model where the variable h_s is 1 if the pattern has a home game in slot s and 0 if it has an away game.

$$\sum_{s=1}^{n-1} h_s \leq n/2 \tag{4.1}$$

$$\sum_{s=1}^{n-1} h_s \geq n/2 - 1 \tag{4.2}$$

$$\sum_{s=\hat{s}}^{\hat{s}+2} h_s \leq 2 \qquad \forall \hat{s} \in \{1, \ldots, n-3\} \tag{4.3}$$

$$\sum_{s=\hat{s}}^{\hat{s}+2} h_s \geq 1 \qquad \forall \hat{s} \in \{1, \ldots, n-3\} \tag{4.4}$$

$$\sum_{s=1}^{n-2} (h_s = h_{s+1}) = B^1 \tag{4.5}$$

$$h_s \in \{0, 1\} \qquad \forall s \in \{1, \ldots, n-1\} \tag{4.6}$$

The constraints (4.1) and (4.2) make sure that the pattern has at most $n/2$ home games and at most $n/2$ away games. Constraints (4.3) and (4.4) limit the number of consecutive home and consecutive away games to be less than or equal to 2, and constraint (4.5) makes sure that the number of breaks equals B^1.

Similarly, the feasible patterns for Part 2 with B^2 breaks correspond to the feasible solutions of the CP model below.

$$\mathit{sequence}(1, 2, 3, [h_n, \ldots, h_{3n-3}], [0, 1], [n-1, n-1]) \tag{4.7}$$

$$\sum_{s=n}^{3n-4} (h_s = h_{s+1}) = B^2 \tag{4.8}$$

$$h_{3n-4} \neq h_{3n-3} \tag{4.9}$$

$$h_s \in \{0, 1\} \qquad \forall s \in \{n, \ldots, 3n-3\} \tag{4.10}$$

Here constraint (4.7) makes sure that the pattern has $n-1$ home games, $n-1$ away games and no more than two consecutive home games or two consecutive away games. Constraint (4.8) determines the number of breaks, and constraint (4.9) avoids a break in the last slot.

In addition to the two sets P^1 and P^2, we let P_i^p denote the set of patterns which satisfy the hard place constraints of team i for each part p and each team $i \in T$. We also need some coefficients to evaluate the patterns in the master problem. For each pattern $j \in P^p$, $p \in \{1, 2\}$, the coefficient c_j^{Br} denotes the break penalty and is equal to the number of breaks B^p times the break coefficient c^{Br}. The coefficient c_{ij}^{Pl}

denotes the place penalty incurred if team i uses pattern j and is equal to the sum of place coefficients c^{Pl} associated with the violated place constraints. Furthermore, for each pattern $j \in P^1$ the coefficient c_j^{Be} denotes the beginning penalty which is equal to the beginning coefficient c^{Be}, if j has a break in the second slot and 0 otherwise. By letting $c_{ij}^1 = c_j^{Br} + c_{ij}^{Pl} + c_j^{Be}$ for each $i \in T$ and $j \in P^1$ and $c_{ij}^2 = c_j^{Br} + c_{ij}^{Pl}$ for each $i \in T$ and $j \in P^2$, we can let c_{ij}^p denote the coefficient of assigning team i to pattern j in Part p. These coefficients are used in the objective function of the master problem.

4.2 Pattern set

When patterns have been generated for both parts we use an IP model to find a pattern set and assign teams to patterns. In the model, the parameters h_{js} for all $j \in P^p$, $s \in S^p$ and $p \in \mathcal{P}$ represent the patterns and h_{js} is 1 if pattern j has a home game in slot s and 0 if it has an away game. We use a binary variable x_{ij}^p for each team $i \in T$, each pattern $j \in P^p$ and each part $p \in \mathcal{P}$ to assign teams to patterns. The variable x_{ij}^p is 1 if team i uses pattern j in Part p, and it is 0 otherwise.

In addition to the assignment variables we use two kinds of penalty variables π_i^{Br} and π_{ls}^{Ge}. The first variable π_i^{Br} is 1 if team i has a break in the first slot in Part 2, and it is 0 otherwise. The second variable π_{ls}^{Ge} is 1 if the geographic constraint l is violated in slot s, and 0 otherwise.

Finally, we need a variable v which is used for optimality cuts to be explained in Section 4.4. This gives the following IP model.

$$\min \sum_{p\in\mathcal{P}} \sum_{i\in T} \sum_{j\in P_i^p} c_{ij}^p x_{ij}^p + \sum_{i\in T} c^{Br}\pi_i^{Br} + \sum_{l\in C^{Ge}} \sum_{s\in S} c^{Ge}\pi_{ls}^{Ge} + v \tag{4.11}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} \sum_{i\in T} \sum_{j\in P_i^p} c_{ij}^p x_{ij}^p + \sum_{i\in T} c^{Br}\pi_i^{Br} + \sum_{l\in C^{Ge}} \sum_{s\in S} c^{Ge}\pi_{ls}^{Ge} + v \geq LB \tag{4.12}$$

$$\sum_{p\in\mathcal{P}} \sum_{i\in T} \sum_{j\in P_i^p} c_{ij}^p x_{ij}^p + \sum_{i\in T} c^{Br}\pi_i^{Br} + \sum_{l\in C^{Ge}} \sum_{s\in S} c^{Ge}\pi_{ls}^{Ge} + v \leq UB \tag{4.13}$$

$$\sum_{j\in P_i^p} x_{ij}^p = 1 \qquad \forall i \in T,\ \forall p \in \mathcal{P} \tag{4.14}$$

$$\sum_{i\in T} x_{ij}^p \leq 1 \qquad \forall j \in P^p,\ \forall p \in \mathcal{P} \tag{4.15}$$

$$\sum_{i\in T} \sum_{j\in P_i^p} h_{js} x_{ij}^p = n/2 \qquad \forall s \in S^p,\ \forall p \in \mathcal{P} \tag{4.16}$$

$$\sum_{i\in T_l^{Ge}} \sum_{j\in P_i^p} h_{js} x_{ij}^p + \pi_{ls}^{Ge} \geq 1 \qquad \forall l \in C_H^{Ge},\ \forall s \in S^p,\ \forall p \in \mathcal{P} \tag{4.17}$$

$$\sum_{i\in T_l^{Ge}} \sum_{j\in P_i^p} (1 - h_{js}) x_{ij}^p + \pi_{ls}^{Ge} \geq 1 \qquad \forall l \in C_A^{Ge},\ \forall s \in S^p,\ \forall p \in \mathcal{P} \tag{4.18}$$

$$\sum_{j\in P_i^1} h_{jn-1} x_{ij}^1 + \sum_{j\in P_i^2} h_{jn} x_{ij}^2 + \pi_i^{Br} \geq 1 \qquad \forall i \in T \tag{4.19}$$

$$\sum_{j \in P_i^1} (1 - h_{jn-1}) x_{ij}^1 + \sum_{j \in P_i^2} (1 - h_{jn}) x_{ij}^2 + \pi_i^{Br} \geq 1 \qquad \forall i \in T \tag{4.20}$$

$$\sum_{j \in P_i^1} (h_{jn-2} + h_{jn-1}) x_{ij}^1 + \sum_{j \in P_i^2} h_{jn} x_{ij}^2 \leq 2 \qquad \forall i \in T \tag{4.21}$$

$$\sum_{j \in P_i^1} (h_{jn-2} + h_{jn-1}) x_{ij}^1 + \sum_{j \in P_i^2} h_{jn} x_{ij}^2 \geq 1 \qquad \forall i \in T \tag{4.22}$$

$$\sum_{j \in P_i^1} h_{jn-1} x_{ij}^1 + \sum_{j \in P_i^2} (h_{jn} + h_{jn+1}) x_{ij}^2 \leq 2 \qquad \forall i \in T \tag{4.23}$$

$$\sum_{j \in P_i^1} h_{jn-1} x_{ij}^1 + \sum_{j \in P_i^2} (h_{jn} + h_{jn+1}) x_{ij}^2 \geq 1 \qquad \forall i \in T \tag{4.24}$$

$$x_{ij}^p \in \{0, 1\} \qquad \forall i \in T, \ \forall j \in P^p, \ \forall p \in \mathcal{P} \tag{4.25}$$

$$\pi_i^{Br} \in \{0, 1\} \qquad \forall i \in T \tag{4.26}$$

$$\pi_{ls}^{Ge} \in \{0, 1\} \qquad \forall l \in C^{Ge}, \ \forall s \in S \tag{4.27}$$

$$v \in \mathbb{R}_+ \tag{4.28}$$

The objective function (4.11) includes break penalties, beginning penalties, place penalties, geographic penalties and the coefficient used for optimality cuts. Constraints (4.12) and (4.13) give upper and lower bounds on the objective value. Constraints (4.14) make sure that all teams are assigned to a pattern in each part, and constraints (4.15) say that a pattern cannot be assigned to more than one team. Constraints (4.16) require that exactly half the teams play home in each slot and constraints (4.17) and (4.18) state the geographic constraints. Constraints (4.19) and (4.20) make sure that π_i^{Br} is 1 if team i has a break in slot n, and the constraints (4.21) – (4.24) are used to avoid more than 2 consecutive home games or more than 2 consecutive away games in the transition between Part 1 and Part 2.

In case the problem is infeasible, we return to Step 1 and generate additional patterns. Otherwise, we let $(\bar{x}, \bar{\pi}^{Br}, \bar{\pi}^{Ge})$ denote an optimal solution to the problem, we let $\bar{P}^p$ denote the set of patterns used in Part p, and in case $LB < \bar{v}$, we update LB to be equal to $\bar{v}$. Finally, we let the parameters $\bar{h}_{is}^1$ for each $s \in S^1$ represent the pattern team i uses in Part 1 and $\bar{h}_{is}^2$ for each $s \in S^2$ represent the pattern team i uses in Part 2.

4.3 Feasibility checks

After having found a pattern set, we need to check feasibility. We start by checking feasibility of Part 1 and Part 2 separately. In case one or both of these are infeasible, cuts are added to the master problem and we return to Step 2. Otherwise we continue with examining feasibility of the combined pattern set for both parts.

4.3.1 Feasibility of Part 1

Miyashiro, Iwasaki and Matsui[12] give a necessary condition for feasibility of a pattern set which can be used to check feasibility of Part 1. The necessary condition gives a lower bound on the number of mutual games which must be scheduled between a subset of patterns. A subset $\hat{P}^1 \subseteq \bar{P}^1$ with cardinality Z must allow $\frac{Z(Z-1)}{2}$ mutual games, since all the teams associated with patterns from $\hat{P}$ must meet. In each slot the number of mutual games cannot exceed the minimum of home games and away games played by teams associated with $\hat{P}^1$ and this leads to the condition

$$\sum_{s \in S^1} \left(\min \left\{ \sum_{j \in \hat{P}^1} h_{js}, \sum_{j \in \hat{P}^1} (1 - h_{js}) \right\} \right) \geq \frac{Z(Z-1)}{2}$$

for all subsets $\hat{P}^1 \subseteq \bar{P}^1$.

Rasmussen and Trick[14] presented a MILP model to check if the condition is satisfied for all subsets of cardinality Z. The model finds the subset of patterns with fewest possible mutual games. The binary variable α_j is 1 if pattern j is included in the subset, and 0 otherwise. The binary variable δ_s is 1 if home games are counted in slot s, and 0 if away games are counted. Finally, the variable β_s counts the number of home or away games in slot s. The model is known as (UBM) and looks as follows:

$$\min \sum_{s \in S^1} \beta_s \tag{4.29}$$

$$\text{s.t.} \sum_{j \in \bar{P}^1} \alpha_j = Z \tag{4.30}$$

$$\beta_s - \sum_{j \in \bar{P}^1} h_{js} \alpha_j + Z(1 - \delta_s) \geq 0 \qquad s \in S^1 \tag{4.31}$$

$$\beta_s - \sum_{j \in \bar{P}^1} (1 - h_{js}) \alpha_j + Z \delta_s \geq 0 \qquad s \in S^1 \tag{4.32}$$

$$\alpha_j, \ \delta_s \in \{0, 1\} \qquad j \in \bar{P}^1, \ s \in S^1 \tag{4.33}$$

$$\beta_s \in \mathbb{R}_+ \qquad s \in S^1 \tag{4.34}$$

The objective function (4.29) minimizes the number of possible mutual games. The constraint (4.30) makes sure that Z teams are included in the subset and constraints (4.31) and (4.32) require that β_s is equal to the number of home games if $\delta_s = 1$ and equal to the number of away games if $\delta_s = 0$.

We can now use the pattern diversity condition[14] to perform the first feasibility check.

> **The pattern diversity condition:** *Given a pattern set $\bar{P}^1$ and a subset size Z, then the pattern set is feasible only if the optimal solution of (UBM) is no less than $\frac{Z(Z-1)}{2}$.*

Notice that the condition was originally stated for a double round robin problem but we have changed the threshold value $Z(Z-1)$ to $\frac{Z(Z-1)}{2}$. If the condition is violated, we add the following logic-based Benders cut to the master problem:

$$\sum_{i \in T} \sum_{j \in \hat{P}^1} x_{ij}^1 \leq Z - 1 \tag{4.35}$$

where $\hat{P}^1 = \{j \in \bar{P}^1 | \alpha_j = 1\}$.

4.3.2 Feasibility of Part 2

To check feasibility of Part 2, we use both the pattern diversity condition and the *multiple pattern separation condition* known from [14]. However, both feasibility checks can be strengthened compared to the original presentation due to the additional constraints considered in this application.

Since all teams must meet once in both halves of Part 2 we can apply the pattern diversity condition on both halves of Part 2 in the same way we applied it on Part 1. This would lead to cuts similar to (4.35). However, we can apply a much stronger cut when we use the fact that multiple patterns in Part 2 can have identical first halves or identical second halves.

Assume that a set of patterns $\hat{P}^2$ cannot meet in the first half of Part 2. This gives us the cut

$$\sum_{i \in T} \sum_{j \in \hat{P}^2} x^2_{ij} \leq Z - 1 \tag{4.36}$$

but instead we add the cut

$$\sum_{i \in T} \sum_{j \in \hat{P}^2_E} x^2_{ij} \leq Z - 1 \tag{4.37}$$

where $\hat{P}^2_E$ is the extended set of patterns consisting of all patterns which have a first half identical to one of the patterns in $\hat{P}^2$. A cut for the last half can be strengthened similarly.

The multiple pattern separation condition checks if the mutual games between the teams using a subset of patterns $\hat{P}^2 \subseteq \bar{P}^2$ can be assigned to slots. The following CP model is used to find a feasible assignment of games to slots if any exists, and otherwise we know that $\hat{P}^2$ is an infeasible subset. The model uses the variable $\sigma_{j_1 j_2}$ and sets it equal to the slot where the team using pattern j_1 plays home against the team using pattern j_2.

$$(h_{j_1 s} = 0) \vee (h_{j_2 s} = 1) \Rightarrow (\sigma_{j_1 j_2} \neq s) \qquad \forall j_1, j_2 \in \hat{P},\ \ j_1 \neq j_2,\ \ \forall s \in S^2 \tag{4.38}$$

$$\mathit{alldifferent}\left(\mathit{all}(j_2 \in \hat{P}^2 \setminus j_1)\sigma_{j_1 j_2},\ \ \mathit{all}(j_2 \in \hat{P}^2 \setminus j_1)\sigma_{j_2 j_1}\right) \qquad \forall j_1 \in \hat{P}^2 \tag{4.39}$$

$$(\sigma_{j_1 j_2} - \sigma_{j_2 j_1} < -k) \vee (\sigma_{j_1 j_2} - \sigma_{j_2 j_1} > k) \qquad \forall j_1, j_2 \in \hat{P}^2,\ \ j_1 < j_2 \tag{4.40}$$

$$(\sigma_{j_1 j_2} \leq 2n - 2) \Leftrightarrow (\sigma_{j_2 j_1} \geq 2n - 1) \qquad \forall j_1, j_2 \in \hat{P}^2,\ \ j_1 < j_2 \tag{4.41}$$

$$\sigma_{j_1 j_2} \in S^2 \qquad \forall j_1, j_2 \in \hat{P}^2,\ \ j_1 \neq j_2 \tag{4.42}$$

The constraints (4.38) make sure that the team using pattern j_1 has a home game and the team using pattern j_2 has an away game when the latter team visits the first. Constraints (4.39) require that all games involving the same pattern are scheduled in different slots and constraints (4.40) state the separation constraints between games with the same two opponents. Finally, the constraints (4.41) state that all pairs of teams must play a game in both halves of Part 2.

> **The multiple pattern separation condition:** *Given a pattern set $\bar{P}^2$ and a subset of patterns $\hat{P}^2$ from this set, then the pattern set is feasible only if the above CP model has a feasible solution.*

If a subset of patterns $\hat{P}^2$ with cardinality Z makes the CP model infeasible, we add the logic Benders cut

$$\sum_{i \in T} \sum_{j \in \hat{P}^2} x_{ij}^2 \leq Z - 1 \tag{4.43}$$

to the master problem.

In [14] all subsets of patterns with cardinality less than a lower bound were checked in order to find infeasible subsets. Instead, we use a heuristic approach for finding candidate subsets since it is faster and it allows us to check subsets with larger cardinality. The price we pay is the risk of missing an infeasible subset with small cardinality.

The idea is to find the subset of teams which are most likely to make the CP model infeasible. We do that by finding a subset of patterns $\hat{P}^2$ which can only play a small number of mutual games and where the games must be played close to the middle of Part 2 since this will conflict with the separation constraints.

For a given cardinality Z, we let $\hat{P}^2$ be equal to the subset of $\bar{P}^2$ which has cardinality Z and minimizes

$$\sum_{j_1 \in \hat{P}^2} \sum_{j_2 \in \hat{P}^2} v_{j_1 j_2}$$

where

$$v_{j_1 j_2} = \sum_{s=n}^{2n-2-k} |\hat{h}_{j_1 s}^2 - \hat{h}_{j_2 s}^2| + \sum_{s=2n-1-k}^{2n-2+k} \frac{1}{2} |\hat{h}_{j_1 s}^2 - \hat{h}_{j_2 s}^2| + \sum_{s=2n-1+k}^{3(n-1)} |\hat{h}_{j_1 s}^2 - \hat{h}_{j_2 s}^2|$$

The parameter $v_{j_1 j_2}$ increases with the number of slots in which j_1 and j_2 can meet, and games close to the middle of Part 2 contribute less than games in the beginning or in the end of Part 2.

The multiple pattern separation condition is used for increasing cardinalities until we find an infeasible subset or until we have checked all cardinalities.

4.3.3 Feasibility of the combined pattern set

We use two kinds of feasibility checks for the combined pattern set. First all pairs of patterns are checked to see if the required number of games can be scheduled without violating the separation constraints. If this is not the case, we add a cut for each pair of teams which violates the constraints.

Assume that we have two patterns which violate the separation constraints and the first pattern consists of patterns j_1^1 and j_1^2 while the second pattern consists of patterns j_2^1 and j_2^2. If the first pattern is assigned to team i_1 and the second to team i_2, we add the following logic-based Benders cut to the master problem.

$$x_{i_1 j_1^1}^1 + x_{i_1 j_1^2}^2 + x_{i_2 j_2^1}^1 + x_{i_2 j_2^2}^2 \leq 3.$$

If all pairs of patterns satisfy the separation constraints, we use the multiple pattern separation condition on the combined pattern set. Again we need a CP model to check feasibility but this time we consider a subset of teams $\hat{T}$, and for each team i, we let j_i^1 and j_i^2 be the patterns assigned to team i in Part 1 and Part 2, respectively. In order to ease notation let $\bar{h}_{is} = \bar{h}^1_{j_i^1 s}$ for all $s \in S^1$ and $\bar{h}_{is} = \bar{h}^2_{j_i^2 s}$ for all $s \in S^2$ in the following CP model. We can then formulate the CP model by letting the variable $\sigma^1_{i_1 i_2}$ denote the slot where the teams i_1 and i_2 meet in Part 1 and by letting $\sigma^2_{i_1 i_2}$ denote the slot where team i_2 visits team i_1 in Part 2.

$$(\bar{h}_{i_1 s} = \bar{h}_{i_2 s}) \Rightarrow (\sigma^1_{i_1 i_2} \neq s) \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 < i_2,\ \ \forall s \in S^1 \tag{4.44}$$

$$\sigma^1_{i_1 i_2} = \sigma^1_{i_2 i_1} \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 < i_2 \tag{4.45}$$

$$\mathit{alldifferent}\left(\mathit{all}(i_2 \in \hat{T} \setminus i_1)\sigma^1_{i_1 i_2}\right) \qquad \forall i_1 \in \hat{T} \tag{4.46}$$

$$\sigma^1_{i_1 i_2} < \sigma^2_{i_1 i_2} - k \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 \neq i_2 \tag{4.47}$$

$$(\bar{h}_{i_1 s} = 0) \vee (\bar{h}_{i_2 s} = 1) \Rightarrow (\sigma^2_{i_1 i_2} \neq s) \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 \neq i_2,\ \ \forall s \in S^2 \tag{4.48}$$

$$\mathit{alldifferent}\left(\mathit{all}(i_2 \in \hat{T} \setminus i_1)\ \sigma^2_{i_1 i_2},\ \ \mathit{all}(i_2 \in \hat{T} \setminus i_1)\sigma^2_{i_2 i_1}\right) \qquad \forall i_1 \in \hat{T} \tag{4.49}$$

$$(\sigma^2_{i_1 i_2} < \sigma^2_{i_2 i_1} - k) \vee (\sigma^2_{i_2 i_1} < \sigma^2_{i_1 i_2} - k) \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 < i_2 \tag{4.50}$$

$$(\sigma^2_{i_1 i_2} \leq 2n-2) \Leftrightarrow (\sigma^2_{i_2 i_1} \geq 2n-1) \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 < i_2 \tag{4.51}$$

$$\sigma^1_{i_1 i_2} \in S^1 \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 \neq i_2 \tag{4.52}$$

$$\sigma^2_{i_1 i_2} \in S^2 \qquad \forall i_1, i_2 \in \hat{T},\ \ i_1 \neq i_2 \tag{4.53}$$

In this model constraints (4.44) make sure that, in Part 1, two teams can only meet in a slot where one of the teams plays home and the other plays away. Constraints (4.45) say that the model does not distinguish between home and away games in Part 1 and constraints (4.46) require that a team does not play more than one game in the same slot in Part 1. Constraints (4.47) make sure that the required separation between games with the same opponents is satisfied between games from Part 1 and Part 2. The constraints (4.48) - (4.51) are similar to the constraints (4.38) - (4.41).

In case a subset of teams $\hat{T}$ makes this CP model infeasible, we add the following logic-based Benders cut to the master problem.

$$\sum_{i \in \hat{T}} (x^1_{ij_i^1} + x^2_{ij_i^2}) \leq 2|\hat{P}| - 1 \tag{4.54}$$

We check subsets of teams with cardinality less than or equal to 4 and return to Step 2 when we find an infeasible subset.

4.4 Timetable

If no cuts have been generated in Step 3, we use an IP model to find an optimal timetable for the given pattern set or to prove that the pattern set is infeasible. In this model we use a binary variable $y_{i_1 i_2 s}$ for all $i_1, i_2 \in T$ and for all $s \in S$. For a slot s in Part 1 the variable $y_{i_1 i_2 s}$ is 1, if the teams i_1 and i_2 meet in slot s, and for slots in Part 2 it is 1, if team i_1 plays home against team i_2 in slot s.

In the model we consider the game, home and top team constraints and for each of these constraints we associate a penalty variable which becomes 1 if the constraint is violated. The penalty variables are denoted π^{Ga}, π^{Ho} and π^{To}, respectively.

Before we state the model, recall that T_l^{Ga} is the pair of teams involved in game constraint l and S_l^{Ga} is the set of slots at which the game must be played. T_l^{Ho} is the pair of teams (i_1, i_2) involved in home constraint l and the team i_1 must play home to satisfy the constraint. To ease notation we use a parameter $\bar{h}_{is}$ for all $s \in S$ to represent the pattern assigned to team i, we let $S^1_{i_1 i_2}$ denote the slots in Part 1 where teams i_1 and i_2 can meet and we let $S^2_{i_1 i_2}$ denote the set of slots where team i_2 can visit team i_1. Furthermore, we let S_k^p denote the set of slots $\{p(n-1)+1-k, \ldots, p(n-1)\}$ for all $p \in \mathcal{P}$, since this set is used in the separation constraints.

$$\min \quad \sum_{i \in T^{To}} c^{To} \pi_i^{To} + \sum_{l \in C^{Ga}} c^{Ga} \pi_l^{Ga} + \sum_{l \in C^{Ho}} c^{Ho} \pi_l^{Ho} \tag{4.55}$$

$$\text{s.t.} \quad y_{i_1 i_2 s} = 0 \qquad \forall i_1, i_2 \in T, i_1 > i_2 \;\; s \in S^1 \tag{4.56}$$

$$\sum_{i_2 \in T \setminus i_1} (y_{i_1 i_2 s} + y_{i_2 i_1 s}) = 1 \qquad \forall i_1 \in T, \;\; \forall s \in S^p, \;\; p \in \mathcal{P} \tag{4.57}$$

$$\sum_{s \in S^1_{i_1 i_2}} y_{i_1 i_2 s} = 1 \qquad \forall i_1, i_2 \in T, \;\; i_1 < i_2 \tag{4.58}$$

$$\sum_{s \in S^2_{i_1 i_2}} y_{i_1 i_2 s} = 1 \qquad \forall i_1, i_2 \in T, \;\; i_1 \neq i_2 \tag{4.59}$$

$$\sum_{s=l(n-1)+1}^{(l+1)(n-1)} (y_{i_1 i_2 s} + y_{i_2 i_1 s}) = 1 \qquad \forall i_1, i_2 \in T, \;\; i_1 < i_2, \;\; \forall l \in \{1, 2\} \tag{4.60}$$

$$\sum_{s=\bar{s}}^{\bar{s}+k} (y_{i_1 i_2 s} + y_{i_2 i_1 s}) \leq 1 \qquad \forall i_1, i_2 \in T, \;\; i_1 < i_2, \;\; \forall \bar{s} \in S_k^p, \;\; \forall p \in \mathcal{P} \tag{4.61}$$

$$\sum_{s \in S_l^{Ga}} (y_{i_1 i_2 s} + y_{i_2 i_1 s}) + \pi_l^{Ga} \geq 1 \qquad (i_1, i_2) = T_l^{Ga}, \;\; \forall l \in C^{Ga} \tag{4.62}$$

$$\sum_{s \in S^1} \bar{h}_{i_1 s} (y_{i_1 i_2 s} + y_{i_2 i_1 s}) + \pi_l^{Ho} \geq 1 \qquad (i_1, i_2) = T_l^{Ho}, \;\; \forall l \in C^{Ho} \tag{4.63}$$

$$\sum_{i_2 \in T \setminus C_{i_1}^{To}} \sum_{s \in S^1} \bar{h}_{i_1 s} (y_{i_1 i_2 s} + y_{i_2 i_1 s}) + \pi_{i_1}^{To} \geq 1 \qquad \forall i_1 \in T^{To} \tag{4.64}$$

$$y_{i_1 i_2 s} \in \{0, 1\} \qquad \forall i_1, i_2 \in T, \;\; s \in S \tag{4.65}$$

$$\pi_i^{To} \in \{0, 1\} \qquad \forall i \in T^{To} \tag{4.66}$$

$$\pi_l^{Ga} \in \{0, 1\} \qquad \forall l \in C^{Ga} \tag{4.67}$$

$$\pi_l^{Ho} \in \{0, 1\} \qquad \forall l \in C^{Ho} \tag{4.68}$$

The objective function (4.55) minimizes the penalties associated with the timetable. The constraints (4.56) require that $y_{i_1 i_2 s}$ is zero if $i_1 > i_2$ for all $s \in S^1$, and constraints (4.57) make sure that all teams play exactly one game in each slot. Constraints (4.58) and (4.59) make sure that all pairs of teams meet once in Part 1 and both teams play a home game against the other team in Part 2. Constraints (4.60) make sure that both halves of Part 2 constitute a single round robin tournament, since they require that all pairs of teams meet in both halves. The separation constraints are satisfied due to constraints (4.61) which require that only one game with the same opponents can be played within $k+1$ consecutive slots. The constraints (4.62), (4.63) and (4.64) require that the penalty variables for the game, home and top constraints are 1 if the corresponding constraints are violated.

In case the model is infeasible, we add the following logic-based Benders cut to the master problem.

$$\sum_{i \in T}(x^1_{ij^1_i} + x^2_{ij^2_i}) < 2n - 1.$$

Otherwise, the optimal solution of the model gives an optimal timetable for the pattern set found in the master problem. This means that we have a feasible schedule and the value of the schedule is the value of the pattern set plus the value of the timetable.

After a feasible schedule has been found, we start searching for a better schedule by setting UB equal to the value of the current best schedule and by adding the following optimality cut to the master problem.

$$v \geq v^{TT}\left(\sum_{i \in T}(x^1_{ij^1_i} + x^2_{ij^2_i}) - 2n\right)$$

where v^{TT} is the value of the timetable which has just been found.

5 Computational results

The performance of the algorithm and the quality of the schedules it obtains have been tested by comparing with the real schedule for the 2005/2006 season and by solving a number of randomly generated instances. The algorithm has been implemented as an OPL Script in Ilog OPL Studio [11] which uses Ilog CPLEX to solve IP problems and Ilog SOLVER to solve CP problems. All the computational tests presented in this section have been performed on a 2.53 GHz Pentium 4 processor with 512 MB RAM and we have used a time limit of 1800 seconds.

The Danish Football Association prefers a k value (separation) of at least 3 but in the schedule for the 2005/2006 season they had to let k equal 0 in order to satisfy team requirements. We have solved the problem with k ranging from 0 to 4 and the algorithm is able to satisfy more team requirements than the real schedule in all instances. We have also tested a *complementary* constraint, since this constraint may be able to help the algorithm in some cases. Two patterns j_1 and j_2 are said to be complementary if $h_{j_1 s} \neq h_{j_2 s}$ for all slots s. The constraint requires that the pattern set consists of pairs of complement patterns only.

The results for the instances with k ranging from 0 to 4 ($k0 - k4$) with and without the complementary constraint (Comp/nComp) are reported in Table 5.1. It states whether a feasible solution has been obtained, whether optimality has been proven, the time used to find the best feasible solution, the time used to prove optimality and the value of the best schedule when all penalty coefficients equal 1.

Although the algorithm has problems with proving optimality we see that it generates very good solutions within a short amount of time and, in practical applications, optimal solutions may not even be the goal. In fact a number of good

Table 5.1: Results for SAS Ligaen.

Instance	Feasible sol. found	Optimality proven	Time to find best solution	Time to prove optimality	Best value
$k0$-nComp	yes	no	707.94	—	41
$k0$-Comp	yes	no	188.11	—	42
$k1$-nComp	yes	no	544.09	—	41
$k1$-Comp	yes	yes	604.67	634.32	41
$k2$-nComp	yes	no	419.05	—	41
$k2$-Comp	yes	no	361.34	—	42
$k3$-nComp	yes	no	458.20	—	42
$k3$-Comp	yes	no	167.55	—	42
$k4$-nComp	yes	no	882.48	—	42
$k4$-Comp	yes	no	1310.81	—	42

Table 5.2: Number of violated constraints.

Instance	Place (29)	Game (1)	Top (10)	Home (1)	Beg. (10)	Geo. (3)	Break (372)	Total (426)
Real schedule	3	0	2	0	2	8	46	61
$k0$-nComp	0	0	1	1	2	1	36	41
$k0$-Comp	0	1	1	0	2	0	38	42
$k1$-nComp	0	0	1	1	2	1	36	41
$k1$-Comp	0	0	1	0	2	0	38	41
$k2$-nComp	0	1	1	0	2	1	36	41
$k2$-Comp	0	0	1	1	2	0	38	42
$k3$-nComp	0	0	1	0	2	1	38	42
$k3$-Comp	0	0	1	1	2	0	38	42
$k4$-nComp	0	0	1	0	2	1	38	42
$k4$-Comp	0	1	1	0	2	0	38	42

feasible schedules may be just as good or even better than one optimal schedule since it can be very hard to determine the right values for the penalty coefficients.

The exact constraints for the 2005/2006 season are confidential but in Table 5.2 we show the number of constraints, the number of violated constraints for the real schedule and the number of violated constraints for each of the 10 instances we have solved.

In all instances we are able to reduce the number of violated constraints with more than 30 percent and at the same time we can increase the separation from 0 to 4. Both top teams have only 5 home games in Part 1 and this means we will always violate at least 1 top team constraint. We also violate 2 beginning constraints in each instance but this is because one of the teams wants to begin with 2 away games. When k is less than 3, the price of using the complementary constraint is 2 additional breaks but, for greater k, there is no difference in the solution values.

In addition to the tests for the 2005/2006 season, we have tested the solution method on 10 randomly generated instances which resembles the real problem. Each of the instances have 30 randomly distributed place constraints, they have 2 top teams, 1 geographic home constraint with three teams, beginning constraints and best half constraints besides the break minimization constraint. Again we have tested the instances with k ranging from 0 to 4, with and without the complementary constraint.

The computational results are displayed in Table 5.3 and show the number of instances in which a feasible schedule has been found, the number of instances in which optimality has been proven, the average time to find the best schedule, the average time to prove optimality and the average value of the best schedule obtained.

Table 5.3: Results for randomly generated instances.

Instance	Feasible solution	Proven optimal	Avg. time to find best solution	Avg. time to prove optimality	Avg. value
k0-nComp	10	10	119.41	185.26	33.50
k0-Comp	10	10	62.56	115.34	33.70
k1-nComp	10	10	136.13	196.47	33.50
k1-Comp	10	10	74.19	139.03	33.70
k2-nComp	10	10	147.70	215.17	33.50
k2-Comp	10	10	98.42	172.70	33.70
k3-nComp	5	3	637.03	859.08	34.20
k3-Comp	10	10	264.37	400.21	33.70
k4-nComp	2	2	995.43	1141.81	33.00
k4-Comp	9	8	561.05	672.84	33.77

The algorithm is very efficient at solving the problems with k less than three but, with k equal to three and four, the problems are getting harder and without the complementary constraint we are only able to solve 5 of the instances with k equal to 3 and 2 of the instances with k equal to 2 within the time limit. On the other hand when the complementary constraint is used, we can solve all instances but 1 and the objective value does not increase significantly. This makes the complementary constraint a good option when hard instances are considered.

6 Conclusion

In this paper we have outlined the problem of finding a seasonal schedule for the Danish soccer league, SAS Ligaen, and presented a solution method to solve the problem. As most sports scheduling applications this is a hard problem, since a large number of constraints are present and these constraints are often conflicting. Many types of constraints have already been considered in the sports scheduling literature but, since this is the first application for a triple round robin tournament, we introduce a number of new constraints.

The solution method uses a logic-based Benders decomposition which decomposes the problem into finding a pattern set and finding a timetable. The computational tests show that good solutions can be obtained in a short amount of time and this makes it possible to generate a number of feasible schedules from which a final schedule can be chosen.

The work has been done in cooperation with the Danish Football Association and after having seen the computational results they have decided to use the algorithm for scheduling the 2006/2007 season.

Although the algorithm works for practical applications, it has problems with proving optimality in most of the instances for the 2005/2006 season. Finding a more efficient way of proving optimality and improving the cuts in general could be a direction for future work in this research area.

7 Acknowledgement

The author would like to thank Benny Olsen, Peter Ebbesen and Allan G. Hansen from the Danish Football Association for valuable discussions of the various constraints and the supply of the constraints for the 2005/2006 season.

References

[1] J. Armstrong, R.J. Willis, Scheduling the Cricket World Cup – a Case Study, Journal of the Operational Research Society 44 (11)(1993)1067–1072.

[2] T. Bartsch, A. Drexl, S. Kröger, Scheduling the professional soccer leagues of Austria and Germany, Computers & Operations Research 33 (7)(2006) 1907–1937.

[3] D. Costa, An evolutionary tabu search algorithm and the NHL scheduling problem, INFOR 33 (3)(1995) 437–461.

[4] F.D. Croce, D. Oliveri, Scheduling the Italian football league: An ILP-based approach, Computers & Operations Research 33 (7)(2006) 1963–1974.

[5] D. de Werra, L. Jacot-Descombes, P. Masson, A Constrained Sports Scheduling Problem, Discrete Applied Mathematics 26 (1990)41–49.

[6] K.K. Easton, Using Integer Programming and Constraint Programming to Solve Sports Scheduling Problems, PhD dissertation, Georgia Institute of Technology (2002).

[7] J.A. Ferland, C. Fleurent, Computer Aided Scheduling for a Sport League, INFOR 29 (1)(1991) 14–24.

[8] C. Fleurent, J.A. Ferland, Allocating games for the NHL using integer programming, Operations Research 41 (4)(1993) 649–654.

[9] M. Henz, Scheduling a major college basketball conference – revisited, Operations Research 49 (2001)163–168.

[10] J.N. Hooker, G. Ottosson, Logic-based Benders decomposition, Mathematical Programming 96 (2003)33–60.

[11] ILOG, ILOG OPL Studio 3.7, Language Manual, 2003.

[12] R. Miyashiro, H. Iwasaki, T. Matsui, Characterizing feasible pattern sets with a minimum number of breaks" in: E. Burke, P. De Causmaecker (Eds.), (PATAT 2002), Lecture Notes in Computer Science 2740, Springer, 2003, pp. 78–99.

[13] G.L. Nemhauser, M.A. Trick, Scheduling a Major College Basketball Conference, Operations Research 46 (1)(1998)1–8.

[14] R.V. Rasmussen, M.A. Trick, A Benders approach for the constrained minimum break problem, European Journal of Operational Research (to appear).

[15] R.A. Russell, J.M.Y. Leung, Devising a Cost Effective Schedule for a Baseball League, Operations Research 42 (4)(1994)614–625.

[16] J.A.M. Schreuder, Combinatorial aspects of construction of competition Dutch Professional Football Leagues, Discrete Applied Mathematics 35 (1992)301–312.