Alcom-FT Technical Report Series
ALCOMFT-TR-01-84

ANGEWANDTE MATHEMATIK UND INFORMATIK
UNIVERSITÄT ZU KÖLN

Report No. 2001-409

# Minimizing Breaks by Maximizing Cuts

by

*Matthias Elf*
*Michael Jünger*
*Giovanni Rinaldi*

30 March 2001

Institut für Informatik
UNIVERSITÄT ZU KÖLN
Pohligstraße 1
D-50969 Köln

Addresses of the authors:

Matthias Elf
Institut für Informatik
Universität zu Köln
Pohligstraße 1
D-50969 Köln
Germany
E-mail: elf@informatik.uni-koeln.de
Telephone: +49-221-470-5314

Michael Jünger
Institut für Informatik
Universität zu Köln
Pohligstraße 1
D-50969 Köln
Germany
E-mail: mjuenger@informatik.uni-koeln.de
Telephone: +49-221-470-5313

Giovanni Rinaldi
Istituto di Analisi dei Sistemi ed Informatica del CNR
Viale Manzoni, 30
I-00185 Roma
Italy
E-mail: rinaldi@iasi.rm.cnr.it
Telephone: +39-6-77161

This work was partially supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT) and by the DONET Consortium of European Research Institutes on Discrete Optimization. DONET is supported by the European Community within the frame of the Training and Mobility of Researchers Programme (contract number ERB TMRX-CT98-0202).

## Abstract

Jean Charles Règin and Michael Trick have proposed to solve the schedule generation problem for sports leagues in two phases in which the first generates a tournament schedule and the second fixes the home-away pattern so as to minimize the number of breaks. While constraint programming techniques appear to be the methods of choice for the first phase, we propose to solve the break minimization problem in sports scheduling by transforming it into a maximum cut problem in an undirected graph and applying a branch-and-cut algorithm. Our approach outperforms previous approaches with constraint programming and integer programming techniques.

# 1. Introduction

During the Seminar "Constraint Programming and Integer Programming", Schloß Dagstuhl, Germany, 17–21 January 2000, the participants tried to identify problems for which a fruitful interaction/competition of constraint programming techniques and integer/combinatorial optimization techniques appears challenging and is likely to enhance the interaction of both communities. One problem area the participants agreed upon consists of various feasibilities/optimization problems occuring in scheduling sports tournaments. The break minimization problem for sports leagues was adressed by both communities during the workshop, in particular by Jean Charles Régin of the constraint programming community and by Michael Trick of the integer programming/combinatorial optimization community.

In section 2, we will define the break minimization problem and give a summary of current solution approaches. In section 3, we will show how the problem can be transformed to a maximum cut problem in an undirected graph with edge weights. In section 4, we will report on computational results on random and real world instances. It will turn out that our approach is applicable to problem instances of size beyond those that could be handled priviously by constraint programming as well as integer programming/combinatorial optimization techniques. In section 5, we will discuss our results and open questions that we hope to resolve in a future version of this paper.

# 2. Break Minimization

We deal with the situation where in a sports league consisting of an even number $n$ of teams each team plays each other team once in $n-1$ consecutive weeks. Each game is played in one of the two opponents home towns, such that the following restrictions apply to each feasible schedule:

[FS1] For each team, the teams played in weeks $1, \ldots, n-1$ are a permutation of all other teams.

[FS2] If in week $w$ team $i$ plays team $j$ "at home" ("+") then team $j$ plays team $i$ in week $w$ in $i$'s town, i.e. "away" ("−").

Figure 1 shows two possible schedules for a league of eight teams. The rows show the game plan for each team, column 1 displays a team, colums $w \in \{2, \ldots, n\}$ show the opponent in week $w-1$, "+", and "−", respectively, indicate if the game is at home or away.

In sports scheduling it is considered undesirable if any team plays two consecutive games either both at home or both away. Such a situation is called a *break*. The schedule of Figure 1(a) imposes 8 breaks whereas the schedule in Figure 1(b) imposes only 6 breaks. In both cases, the number of breaks is minimum given the schedule without home-away assignment. Schreuder [Sch92] has shown that, for an even number $n$ of teams, it is always possible to construct a schedule with $n-2$ breaks and he has given an efficient algorithm

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 : | +2 | −3 | −4 | +5 | +6 | −7 | +8 |
| 2 : | −1 | +7 | −5 | +4 | −3 | −8 | +6 |
| 3 : | −7 | +1 | +8 | −6 | +2 | +5 | −4 |
| 4 : | +5 | −8 | +1 | −2 | +7 | −6 | +3 |
| 5 : | −4 | −6 | +2 | −1 | +8 | −3 | +7 |
| 6 : | +8 | +5 | −7 | +3 | −1 | +4 | −2 |
| 7 : | +3 | −2 | +6 | −8 | −4 | +1 | −5 |
| 8 : | −6 | +4 | −3 | +7 | −5 | +2 | −1 |

$(a)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 : | +8 | +3 | −5 | +7 | −2 | +4 | −6 |
| 2 : | +7 | −8 | +4 | −6 | +1 | −3 | +5 |
| 3 : | +6 | −1 | +8 | +5 | −7 | +2 | −4 |
| 4 : | −5 | +7 | −2 | −8 | +6 | −1 | +3 |
| 5 : | +4 | −6 | +1 | −3 | +8 | +7 | −2 |
| 6 : | −3 | +5 | −7 | +2 | −4 | −8 | +1 |
| 7 : | −2 | −4 | +6 | −1 | +3 | −5 | +8 |
| 8 : | −1 | +2 | −3 | +4 | −5 | +6 | −7 |

$(b)$

**Figure 1.** Two feasible schedules for eight teams

to compute such a schedule (along with the home-away assignment). However, sport tournament schedules are subject to a number of requirements, among them restrictions such as “geographically close teams should not play at home during the same week”. Some authors (like Schreuder [Sch92]) propose to start with an optimal schedule with $n-2$ breaks and incorporate the additional requirements at the cost of more breaks, others (like Régin [Reg98,00] and Trick [Tri00]) propose to consider a schedule without home-away assigment that obeys the various (often not formally describable) side conditions and compute a home-away assigment as to minimize the number of breaks. It is the latter attitude we take here: We are given a feasible tournament schedule without home-away assigment and our task is to find a feasible home-away assigment that minimizes the number of breaks.

Régin formulated a constraint programming model with 0-1-variables and was able to solve instances up to size 20. Trick introduced an integer programming formulation and was able to solve instances up to size 22.

The complexity status of the break minimization problem has not yet been determined (to the best of our knowledge), we believe it is NP-hard (see section 5).

# 3. From Minimizing Breaks to Maximizing Cuts

Given a feasible tournament schedule without home-away assigment

$$\begin{array}{rcccc} 1: & t_{11} & t_{12} & \dots & t_{1,n-1} \\ 2: & t_{21} & t_{22} & \dots & t_{2,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n: & t_{n1} & t_{n2} & \dots & t_{n,n-1} \end{array}$$

in which $t_{ij} \in \{1, 2, \dots, n\}$ is the opponent of team $i$ in week $j$, we construct an undirected graph $G = (V, E)$ with a node $v = (i, j) \in V$ for $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, n-1\}$. There is an edge in $E$ between nodes $(i, j)$ and $(k, l)$ in $V$ if and only if $i = k$ and $1 \leq j = l - 1 \leq n - 1$. I.e., $G = (V, E)$ is as follows:

$$\begin{array}{ccccccc} \boxed{(1,1)} & \text{———} & \boxed{(1,2)} & \text{———} & \cdots & \text{———} & \boxed{(1,n-1)} \\ \boxed{(2,1)} & \text{———} & \boxed{(2,2)} & \text{———} & \cdots & \text{———} & \boxed{(2,n-1)} \\ \vdots & & \vdots & & \ddots & & \vdots \\ \boxed{(n,1)} & \text{———} & \boxed{(n,2)} & \text{———} & \cdots & \text{———} & \boxed{(n,n-1)} \end{array}$$

A cut in $G = (V, E)$ i.e. an edge set $C$ of the form $C = \delta(W)$, $W \subseteq V$, where $\delta(W) = \{e \in E \mid v \in W,\ w \in V \setminus W\}$, partitions $V$ into $V = V^+ \cup V^-$ $(V^+ \cap V^- = \emptyset)$, where $V^+$ and $V^-$ are called the different *shores* of the cut. Any home-away assigment corresponds to a cut in $G$ such that, by [FS2], $(i, j) \in V^+$ if and only if $(t_{ij}, j) \in V^-$ and $|E| - |C|$ is the number of breaks. This condition is modelled by stipulating that $(i, j)$ and $(t_{ij}, j)$ belong to different shores of the cut. We could model this by assigning a capacity of 1 to all edges of $G$ introduced so far and a value of $M \geq n(n-2)+1$ to all $\frac{n(n-1)}{2}$ such pairs of nodes: Figure 2 shows the graph $G$ resulting from this transformation for the instance given in Figure 1(a).

The choice of $M$ would trivially guarantee a correct model. Namely, if the maximum capacity cut with these edge weights has value $\Delta$, then assigning "+" to $t_{ij}$ with $(i, j) \in V^+$ and "−" to $t_{ij}$ with $(i, j) \in V^-$ gives a schedule with the minimum number of $\Delta - \frac{n(n-1)}{2}M$ breaks. But, due to a result of Barahona and Mahjoub [BM86] we can do much better. In a later version of this draft, we will work this out, but the bottom line that we need here is that if we switch the signs of the capacities of all edges in the star of any node $v$ in $G$, the maximum capacity cut induces a maximum capacity cut with the original capacities in which $v$ changes shores and all other nodes stay at their shore.

Therefore, for each edge with capacity $M$ we switch the capacity of the star of one of its endnodes. The resulting edge with capacity $-M$ will be in no maximum cut, so we can contract the edge. We obtain a maximum cut instance with $\frac{n(n-1)}{2}$ nodes and $n(n-2)$ edges with capacities either 1 or $-1$. The result for our example is shown in Figure 3,

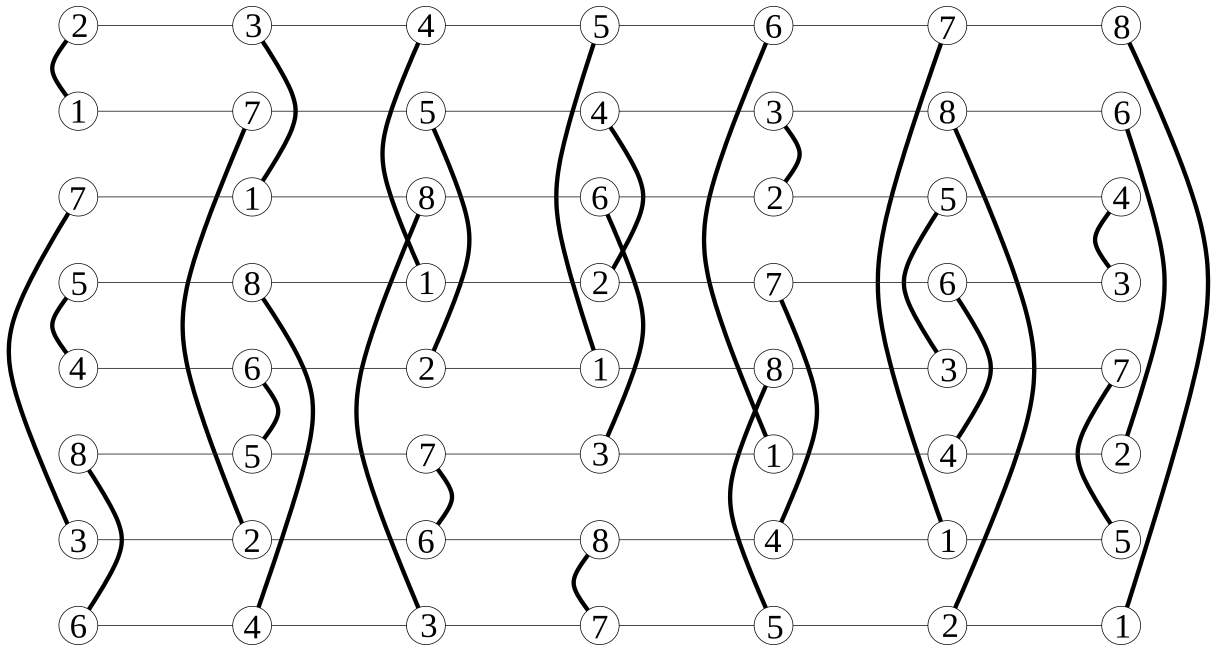



**Figure 2.** Result of the big-$M$ transformation

in which we have (arbitrarily) chosen to switch the cut of the lower indexed vertex each time.)

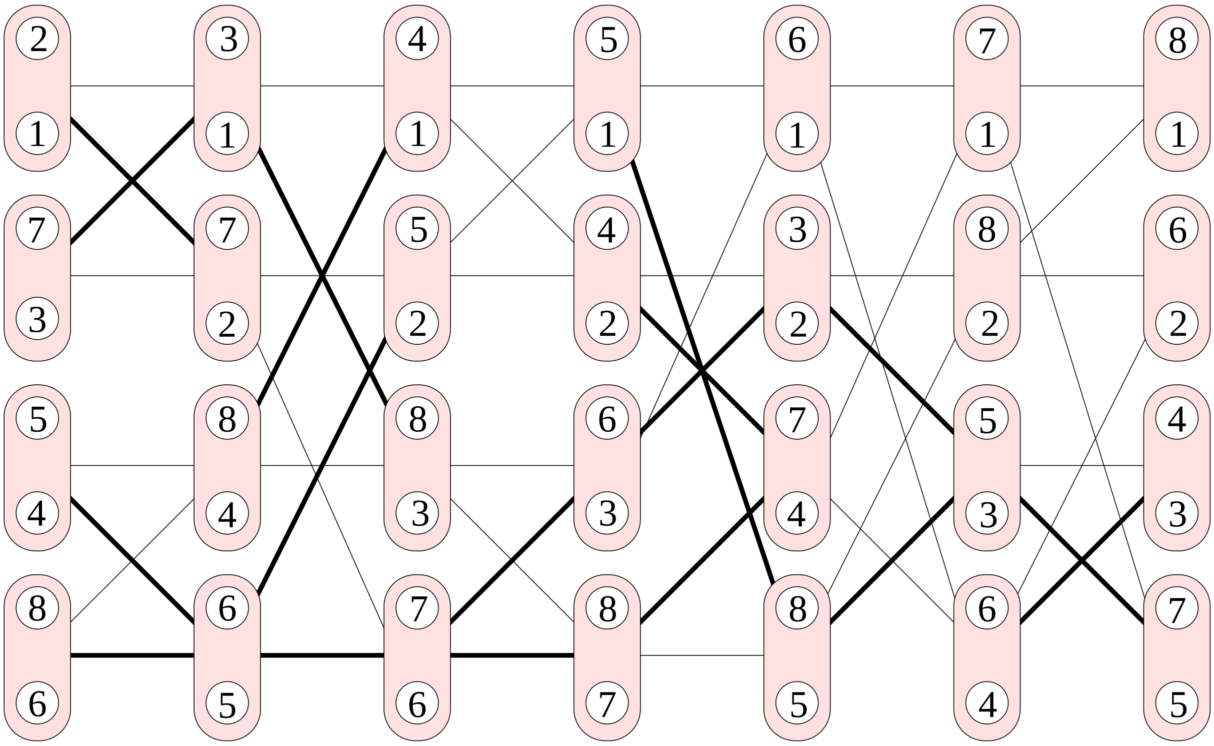
The same graph is displayed in Figure 4 along with a cut of maximum capacity 22 that is indicated by white and grey nodes, respectively.

By backward transformation, this cut corresponds to the home-away assigment displayed in Figure 1(b) with 8 breaks, which proves our claim in section 1 that 8 breaks is minimum for this instance.

# 4. Computational Results

For the computation of maximum capacity cuts we have used an implementation of the algorithm described in [BGJR88] that was reimplemented by Martin Diehl using the ABACUS software [JT00], version 2.3 using CPLEX 6.5 as an LP-Solver. The same implementation was used successfully in, e.g., [DDJ+96] for computing ground states of Ising spin glasses.

The instances were created by computing optimal (($n-2$)-breaks schedules) by Schreuder's procedure and permuting the columns randomly. For each site, we created five random schedules and applied the branch-and-cut algorithm. The results displayed in Table 1 were obtained on a 296MHz Sun Ultra SPARC machine.




*thin edges have capacity 1, fat edges have capacity* $-1$

**Figure 3.** Result of the transformation

We also solved one real world instance, namely the current Bundesliga (first national German soccer league) instance. There are 18 teams and we found that the schedule is already optimal at 16 breaks!

# 5. Discussion

We do not have access to the instances used in the computational studies by Régin and Trick, however, at the time of writing this, it seems that Trick's approach is sligthly ahead in solving 22 teams instances in about 1 hour on a 266 MHz machine. Apparently, our approach is able to handle larger instances easily. (We plan to ask both authors for their instances in order to make direct comparisons.) There are three questions we would like to answer before making this draft into a paper:

1. What is the complexity of the break minimization problem? (We conjecture it is $NP$-hard.)
2. When we give optimal ($(n-2)$-break instances) to our algorithm, the answer $n-2$ is produced very quickly. We would like to establish why this is the case. (We conjecture this is because the corresponding maximum cut instances feature graphs that are not contractible to $K_5$, a class of graphs for which Barahona [Bar83] showed that the maximum cut problem is solvable in polynomial time.)

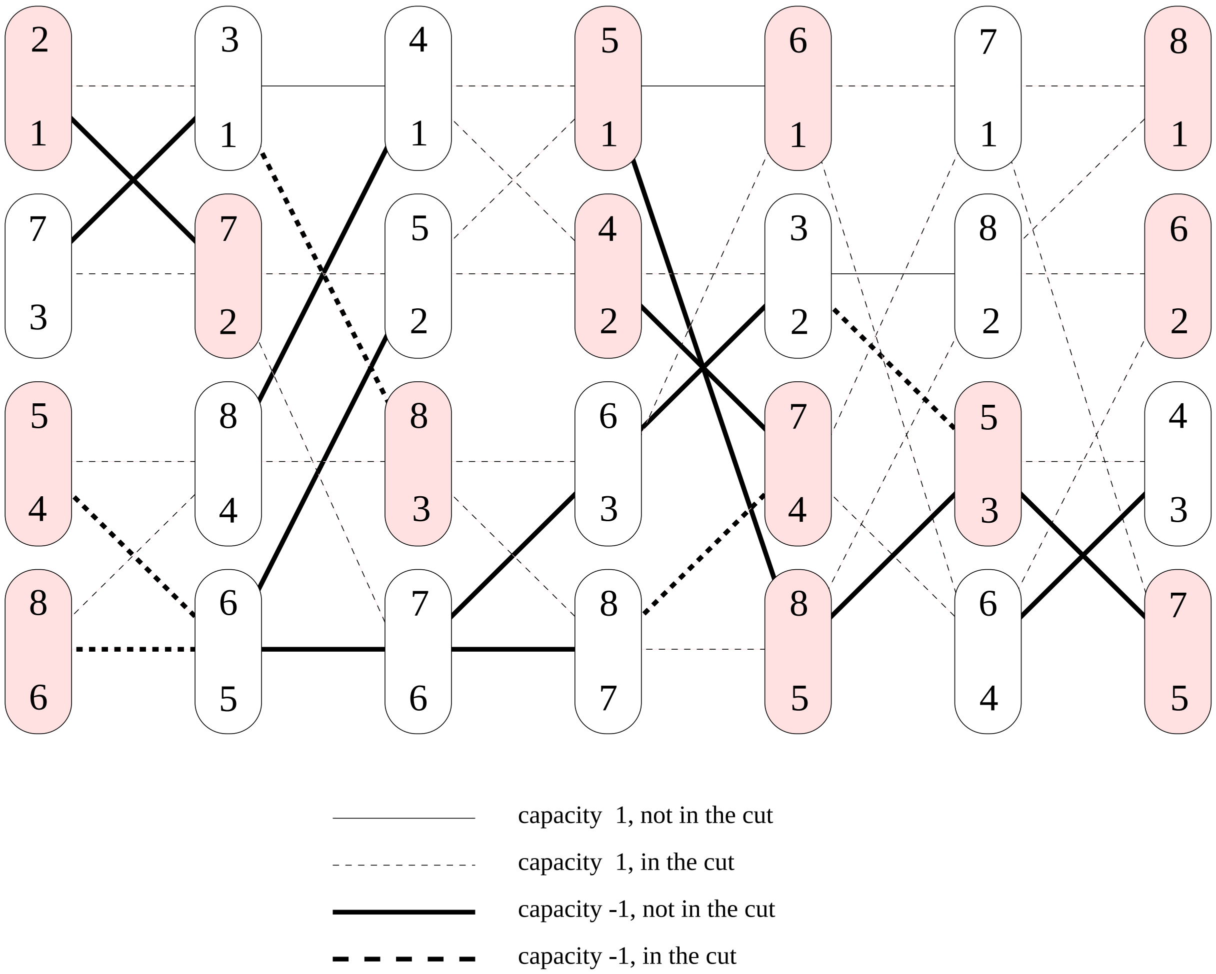



**Figure 4.** A cut of maximum capacity 22

3. What is a random schedule? Is our approach to the random instance generation justified?

# Acknowledgement

We would like to thank Michael Trick for teaching us the break minimization problem and sharing his insights with us.

# References

[Bar83] F. Barahona (1983), "The max-cut problem on graphs not contractible to $K_5$", *Operations Research Letters* **2**, 107–111.

| size | time | | | number of breaks | | |
|---|---|---|---|---|---|---|
| $n$ | minimum | average | maximum | minimum | average | maximum |
| 4 | 0.0 | 0.0 | 0.0 | 2 | 2.0 | 2 |
| 6 | 0.0 | 0.0 | 0.0 | 4 | 4.0 | 4 |
| 8 | 0.0 | 0.0 | 0.0 | 8 | 8.0 | 8 |
| 10 | 0.0 | 0.0 | 0.0 | 10 | 12.2 | 12 |
| 12 | 0.1 | 0.1 | 0.1 | 14 | 16.8 | 18 |
| 14 | 0.1 | 0.3 | 0.5 | 18 | 23.6 | 26 |
| 16 | 0.4 | 1.0 | 2.8 | 28 | 31.2 | 32 |
| 18 | 2.8 | 6.1 | 19.0 | 36 | 40.8 | 44 |
| 20 | 1.1 | 8.7 | 18.5 | 44 | 51.2 | 54 |
| 22 | 6.8 | 36.7 | 94.0 | 58 | 61.2 | 64 |
| 24 | 23.5 | 72.8 | 140.6 | 68 | 72.4 | 74 |
| 26 | 22.9 | 339.0 | 1215.9 | 80 | 85.2 | 90 |

**Table 1.** Computational Results


[BM86] F. Barahona and A. R. Mahjoub (1986), "On the cut polytope", *Mathematical Programming* **36**, 157–173.

[BGJR88] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt (1988), "An application of combinatorial optimization to statistical physics and circuit layout design", *Operations Research* **36**, 493–513.

[DDJ+96] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi (1986), "Exact ground states of 2-dimensional $\pm J$ Ising spin glasses", *Journal of Statistical Physics* **84**, 1363–1371.

[Hen99] M. Henz (1999), "Scheduling a major college basketball conference—revisited", Technical Note, School of Computing, National University of Singapore.

[JT00] M. Jünger and S. Thienel (2000), "The ABACUS System for Branch and Cut and Price Algorithms in Integer Programming and Combinatorial Optimization", to appear in: *Software Practice and Experience*, 2000.

[NT98] G. L. Nemhauser and M. A. Trick (1998), "Scheduling a major college basketball conference", *Operations Research* **46**, 1–8.

[Reg98] J.-C. Régin (1998), "Minimization of the number of breaks in sports scheduling problems using constraint programming", Talk presented at DIMACS Workshop on Constraint Programming and Large Scale Discrete Optimization, Sep. 14–19, 1998.

[Reg00] J.-C. Régin (2000), "Modelling with constraint programming", Talk presented at Dagstuhl Seminar on Constraint Programming and Integer Programming, Jan. 17–21, 2000.

[Sch92] J. A. M. Schreuder (1992), "Combinatorial aspects of construction of competition Dutch Professional Football Leagues", *Discrete Applied Mathematics* **35**, 301–312.

[Tri00] M. A. Trick (2000), "A schedule-then-break approach to sports timetabling", Technical Report, Graduate School of Industrial Administration, Carnegie Mellon University.