**Scheduling a Major College Basketball Conference**

Published in “*Operations Research*” February 1997, Revised May, 1997

George L. Nemhauser
Georgia Institute of Technology

Michael A. Trick
Carnegie Mellon University

**Introduction:**

The nine universities in the Atlantic Coast Conference (ACC) have a basketball competition in which each school plays home and away games against each other over a nine-week period. The ACC and its universities earned in excess of $30 million in basketball revenue in 1995. Almost all of this revenue came from television and radio networks showing the games, and from gate receipts. The creation of a suitable schedule is a very difficult problem with innumerable conflicting requirements and preferences. Besides, some changes made in the last moment or the possibility of adding another team to the Conference are some of the problems the algorithm should include. This paper uses a combination of integer programming and enumerative techniques to approach such a kind of scheduling problem.

**Method used for Scheduling:**

The search for a good schedule can be divided into three steps:

1) A pattern is a string consisting of H (home), A (away), and B (bye), of length equal to the number of slots in the schedule. In step 1, we find a set of patterns with cardinality equal to the number of teams. This set is called a pattern set.

2) In step 2, we assign games to the pattern set consistent with the HAB letters. The result from this stage is a timetable.

3) In step 3, we assign the teams to the patterns. Together with the timetable, this gives a schedule.

Any round robin schedule can be extended to a double round robin by mirroring. In a mirrored schedule, a round robin schedule is repeated, reversing the home and away teams. Various methods for scheduling like doing by hand, constraint programming have been used to get an optimal schedule. The main problem with this approach is the difficulty in finding a solution to steps 1 and 2. For determining these feasible timetables, combinatorial design has been used, but there is no known characterization of all feasible timetables. Furthermore, finding feasible timetables to these problems is so painstaking that very few are known for any given problem size.

**Constraints Involved:**

There are various requirement needed by the teams. There are pattern, game count, team pairing, team requirement, opponent ordering and other constraints. There are considerations given for games played during weekends or weekdays, preference of television networks, final weekend season and most important “Rival Games”.
Integer programming does the process of finding feasible solutions to each of these parts in the algorithm. All feasible patterns are generated and then using enumeration we remove the infeasible ones (Box A). Sets of nine patterns that have a reasonable chance of giving feasible timetables (Box B), are found by using integer programming. The decision variable for each pattern is one if it is in the set. Feasible pattern sets are noted down. The next problem is of getting timetables. An integer program is used to get feasible solutions again. This integer program has an arbitrary objective (Box C), since teams with patterns have not been identified. 234 constraints and approx 300 variables are there in the program. 826 feasible timetables are generated from 17 patterns. Now the next step is to take the timetable and assign the teams to patterns in the pattern set. This process is done through enumeration (Box D). There are 362880 assignments of teams to patterns for each timetable, which are checked for feasibility. All possible 299738880 assignments are looked at and it takes around 24 hrs, giving 17 feasible schedules. Finally these schedules are reduced to 3, considering dominance aspects. It is then submitted to the ACC. The ACC, after consultation with their television partners, choose the official schedule. (Box E). The accepted schedule was actually used for the 1997-1998 season.

Possible Extensions:

There are a number of interesting questions and extensions to this paper. Opportunities to use cost coefficients in the program can be considered. The possibility of using the same algorithm to larger leagues, possibility of replacing the enumerative process and integer programs by efficient algorithms so as to reduce the computational effort used at each of the steps is some other questions are the ones, which come in mind.

Conclusion:

Due to the advent of some problems, which come at the last moment, there is always a possibility that the current schedule has to be changed. Hence we should have a system that can accommodate these changes, update the requirements and generate new schedules quickly. The system here discussed is capable of taking these changes into account and is successful in giving new schedules at good speed. The paper has successfully showed us that by using a suitable combination of enumeration and integer programming, feasible solutions can be attained. The old tiresome and difficult combinatorial design techniques need not be used now. Creating real schedules, meeting and exceeding the requirements of a real college basketball conference show the success of this approach. We can use this approach for solving leagues with 10-11 teams, but for much larger leagues, a heuristic approach will have to be used.

By: Shailesh A. Bangale