

CS182: NBA Scheduler

Abhishek Malali (abhishekmalali@g.harvard.edu)

Charles Liu (cliu02@g.harvard.edu)

Samuel Daulton (sdaulton@g.harvard.edu)

December 7, 2015

1 Introduction

The National Basketball Association is a 30 team league that organizes an 82 game schedule for each team over a 6-month period. It showcases some of the best athletes in the world and has generated globally recognized brands. As such, it is a multi-billion dollar industry and is constantly looking for ways to improve. Recently, one such way has been to reorganize its schedule.

1.1 NBA Scheduling Formula

Currently there are two conferences (East, West) each comprised of three divisions each of which contain five teams. The 82-game schedule is currently set up as follows:

1. 4 games against the other 4 division opponents, $[4 \times 4 = 16 \text{ games}]$
2. 4 games against 6 (out-of-division) conference opponents, $[4 \times 6 = 24 \text{ games}]$
3. 3 games against the remaining 4 conference teams, $[3 \times 4 = 12 \text{ games}]$
4. 2 games against teams in the opposing conference. $[2 \times 15 = 30 \text{ games}]$

And of course, each team must play 41 home and 41 away games.

1.2 Court Availability

As these are actual games with required venues, each team must provide the following:

1. At least 50 dates on which their home court will be available
2. 4 Mondays
3. 4 Thursdays (to help TNT plan its telecasts).

1.3 Official Breaks

There are no games on the following days:

1. Christmas Eve
2. All-Star Week
3. NCAA Men's Division I Basketball Championship Game

1.4 Additional Assumptions

There are no non-NBA events in the venues that may further constrain a team's schedule.

2 Project Goals

While the NBA has obviously been able to put out a schedule every year for every team, there has been recent criticism; the main criticism is there are too many "back-to-back" stretches, where teams potentially play 4 games in 5 nights. Moreover, the length of travel in a short amount of time is too taxing on the players. Our end-goal would be to minimize these schedule-induced "costs" on a player's energy level.

2.1 Data Generation

For the data - we know what the 30 teams are and will calculate the exact distance between any two team venues. Then, for a specific year we will randomly generate the court availability constraints mentioned above as well as the week of the All-Star break (this must be a week in February) and NCAA Championship Game (First Monday of April).

2.2 Initial Schedule

We will then model this problem as a CSP, similar to ones we've seen in class, and generate an initial solution. We will also implement some local search algorithms and test how feasible they are given the number of constraints.

2.3 More Player Friendly Schedule

We will then look to address the player concerns that there are too many games in too short a time. We'll add a new constraint to model this concern: using generic variables N and M , we would add the constraint that a team can play at most N games in M nights. If time permits, we want to find the solution that would minimize the total distance traveled - this lessens the wear on players and saves flight coordination issues. This could potentially be an increasingly computationally intensive problem - so we will look to see if certain optimizations can provide a solution:

1. Some form of greedy next-best search in choosing the next game - choose the next game for a team that makes sure the N games in M nights constraint is met then perform arc consistency
2. The minimizing distance problem for a specific team is similar to traveling salesman problem - try to model this problem in form of A* and find a heuristic based on number of divisional games remaining (divisional games are regionally closer) - then rearrange schedule to fit that team's path. The goal is to minimize the combined distance for every team traveled which will require outside research to see if this is even feasible. At a minimum we will try to beat the total distance traveled from the actual NBA schedule for this year through a greedy approach of finding minimum distance traveled while optimizing for 1 team, then minimum distance traveled for 2 teams starting with the best from the 1-team, and so on until as many teams as possible have minimized their travel distance.
3. Research online for other techniques that haven't been discussed in class - surprisingly this has been an issue well researched in the NFL but not the NBA (probably because that's a smaller problem with only 16 games), but perhaps we can gain some insights from reading papers optimizing their schedule

3 Group Member Roles

The project steps are closely related, so each member's roles will certainly overlap. However, a list of requirements would be:

1. Data generation
2. Initial model formulation
3. Initial CSP search code
4. Local Search code
5. Heuristic Function on travel
6. Model revisited including new constraints
7. Find relevant research papers for NFL

4 Integer Linear Programming

Integer linear programming is a mathematical optimization where all variables are restricted to integers. The objective function and the constraints are linear. In the NBA scheduling problem, the objective function is not defined and the solver has to solve the constraints are formulated in the problem. We solved this problem in 2 steps. The first was to attempt getting a schedule roundwise without considering the dates. Next we included dates and were able to create a player friendly schedule where there no more than 3 games in 5 days.

4.1 Methodology for the roundwise schedule

We create a variable for every team against every other team, for every round. The first part of the variable is the home team followed by the away team and the round number. We introduce the following constraints to the ILP

1. All teams play 82 games , 41 home and away
2. All constraints for the division, in conference and out of conference teams
3. No team can play more than one game in a round
4. Team can only play either home or away

The above ILP was solved using the PuLP as well as the GUROBI solver. The PuLP solver took 40 minutes to converge to solution whereas the GUROBI solver took 6 seconds on an average. The schedule was verified using tests to check if all constraints were satisfied.

4.2 Generating the player friendly schedule

Using the dates generated by datagen, we formulated the ILP for a player friendly schedule. The variables are created for every date instead of every round. To reduce the number of variables we generate the variables for every possible home and away dates only. This reduces the number of variables by about 70%. The time taken to create the constraints and to solve the ILP are greatly reduced. We use the following constraints.

1. All teams play 82 games , 41 home and away
2. All constraints for the division, in conference and out of conference teams
3. No team can play more than one game on a single date
4. Team can only play either home or away
5. A team can only play 3 or lesser matches in 5 days
6. A team does not play 2 away games without a day's break
7. A team does not play an away game right after a home game

The above ILP was solved using the PuLP as well as the GUROBI solver. The PuLP solver took 1 hour to converge to solution whereas the GUROBI solver took 29 seconds on an average. In some cases the 4 games in 5 days was unavoidable due to mismatch in dates. But our solution has minimized it to a great extent and can ensure player vitality.