CS182: NBA Scheduler

Abhishek Malali, Charles Liu (cliu02@g.harvard.edu), Samuel Dalton

October 28, 2015

1 Introduction

The National Basketball Association is a 30 team league that plays an 82 game schedule 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. One such way recently has been to reorganize it's schedule.

1.1 NBA Scheduling Formula

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

1. 4 games against the other 4 division opponents, [4x4=16 games]
2. 4 games against 6 (out-of-division) conference opponents, [4x6=24 games]
3. 3 games against the remaining 4 conference teams, [3x4=12 games]
4. 2 games against teams in the opposing conference. [2x15=30 games]

And of course, each team must play 42 home and 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

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 - mainly that there are too many "back-to-back" stretches, where teams potentially play 4 games in 5 nights, and the length of travel in a short amount of time is too taxing on the players. Our end-goal would be to address these concerns.

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 have generic variables N and M and add the following constraint that no team can play N games in M nights. Additionally, we want to find the solution that would minimize the travel distance of a team - 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 this constraint is met then perform arc consistency

2. Generate some heuristic based on the number of games left against divisional opponents (who would be geographically closer) to perform some kind of A* search

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