The NBA Scheduling Problem

Zhiying Yao Hongyi Ding

2017

December

1 Introduction/Background

The National Basketball Association (NBA) is a men's professional basketball league in North America; composed of 30 teams (29 in the United States and 1 in Canada). It is widely considered to be the premier men's professional basketball league in the world. The NBA is one of the four major professional sports leagues in the United States and Canada. NBA players are the world's best paid athletes by average annual salary per player.

The current league organization divides thirty teams into two conferences of three divisions with five teams each. The central governance organization acts independently with regards to aspects such as bylaw enforcement and season scheduling.

The NBA is a extremely lucrative league, with 5.87 billion in revenue for 2016. In 2016, TV contract were work 2.7 billion dollars. And merchandise accounts for over a billion dollars, with jersey sales alone topping 900 million. Therefore, the schedule of each NBA season significantly affects each team's success, as well as the league's overall revenue.

In this paper, we aim to understand the current NBA scheduling problem, from objective to constraints. Then we will try to solve the problem by formulating and implementing our own model using methods learned in Operations Research, to create a potentially more "optimal" solution.

2 Current NBA Scheduling Algorithm

Until this decade, the NBA had scheduled the entire season completely by hand. The organization is very secretive with regards to the specific method and algorithm for scheduling games.

However, multiple sources confirm that the organization does employ an integer programming method, albeit not completely. The scheduling team at the NBA first hard-codes the marquee games (slots) of the year, such as the opening night games and the Thanksgiving night games. Then, the team uses computers to generate many possible schedules satisfying the "hard-constraints". After generating a list of possible schedules, the team assesses the strengths and weaknesses of each individual schedule. It is here that various "soft-constraints", including "fairness", are examined.

3 Initial Attempt

Our initial attempt is to use a strict integer-programming method to generate our schedule, believing that if the correct values are assigned to each team, the nature of the algorithm would assign the most valuable games to the most valuable slots, as to maximize overall revenue.

Variable Representation

1. Each team has a corresponding triplet representation (i, j, k) as well as a numerical one. Team (i, j, k) represents conference $i \in \{0, 1\}$ (Western = 0, Eastern = 1), division $j \in \{0, 1, 2\}$(3 divisions in each conference), Rank $k \in \{0, 1, 2, 3, 4\}$ (index into each division).

2. There are 24 weeks in the regular NBA season, with 7 slots in each week. Each slot has a duplet representation (l, m), with $k \in (0, 23)$ and $l \in \{0, 4\}$ corresponding to the week number and time slot, respectively. For example, time slot $(14, 3)$ would correspond to the Wednesday slot of week 15.

3. $x_{ijki'j'k'lm}$ is the indicator random variable, representing whether team ijk (the home team) is playing against team $i'j'k'$ (the away team) at week l time slot m.

And our object is to maximize the overall revenue for the league subject to the following constraints.

1. Each team has at most 1 game per day, 4 games per week.

2. Each team has 41 home games and 41 away games throughout the season.

3. Each team has to play 4 games against the other 4 division opponents $(4 \times 4 = 16$ games), 4 games against 6 (out-of-division) conference opponents $(4 \times 6 = 24$ games), 3 games against the remaining 4 conference teams $(3 \times 4 = 12$ games), 2 games against teams in the opposing conference $(2 \times 15 = 30$ games)

4 Another Attempt

However, the integer programming method simply involves too many variables and constraints that the problem cannot be solved within a reasonable amount of time. Thus, we look for an alternative approach to the problem. One thing we notice about the problem is that the problem is very similar to the traveling salesman problem. And one approach to the traveling salesman problem is the hill climbing algorithm. Thus we use the hill climbing algorithm as our second attempt.

In numerical analysis, hill climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found. In the case of NBA scheduling, we use the 2016-2017 schedule as our starting point, which can be downloaded from the official NBA website. Then at each iteration, we randomly switch two games/dates, and check if the new schedule satisfies all our constraints. If so, we evaluate the 'new' schedule, if the new schedule has a better performance than the old one, the change is made to the solution. We repeat this process until either the result converges or it reaches the maximum number of iterations.

The most important part of the algorithm is how we evaluate a schedule. This part is highly objective, since there are many criteria for determining the best schedule, such as total revenue, total distance traveled, and fairness, etc. And due to its various constraints, it is unlikely that we have a perfect schedule that performs very well in each of the category listed above. Therefore, we only consider the potential total revenue of a schedule. Specifically, we are focusing only on the TV-revenue, as this part accounts for a majority of revenue for the NBA.

One of the most important factors in determining the value of a game is the winning record of the two participating teams. People are more likely to watch games between top teams than ones between teams performing poorly.

Of course the winning record alone is not sufficient. Another important factor is the popularity of the team. Although a team with a better winning record tend to be more popular, the opposite is not necessarily true. Sometimes fans love a team because some of its players are very popular, despite the fact the team is not performing well. Therefore, we treat winning record and popularity as two different factors.

In addition to the two factors mentioned above, teams located in big cities such as New York, Chicago, and Los Angeles often have more valuable games in the eyes of a broadcaster, despite not having the best record. This is due to the larger market and fan base of these high net-worth teams.

The last critical factor in determining the value of any game is the time slot assigned. The price of a commercial at a better slot costs more. We find out that the most valuable time slot is the Christmas night slot. And a slot on weekends is much valuable than a slot on weekdays.

Having found the contributing factors for determining the value of a team, what logically follows is the assignment of weights to accurately account for all these factors. This part is highly objective. Fortunately, multiple papers and sports analysts believe that for such a problem, the winning record is twice as important as the popularity of a team. Thus, for the value of a particular team, we assign a 60% weight to the winning record, a 30% weight to the popularity of the team (measured in fan base), and a 10% weight to the location (measured in the market value of the team). And the value of a game is the sum of the values of two participating teams, multiplied by the relative value of the slot. And eventually, the total revenue is the sum of the values of all games.

Here is the pseudocode for our algorithm. The entire algorithm and the results can be found in the link in appendix.

```
s = schedule of previous season
v = evaluator(s) // return a score of the schedule
repeat:
    randomly switch 2 games/dates, get new schedule s'
    check if s' meets all constraints:
        v' = evaluator(s')
```

```
        if v' > v:
            s = s'
output s and v
```

5 Discussion of Results and Solution

Our solution seems to conform with a sensible NBA schedule. For example, high-profile matchups between popular or high-ranked teams such as Golden State Warriors and Cleveland Cavaliers have the prime-time slot of Christmas night, fitting with our objective of maximizing overall revenue.

We compare our generated schedule to the one NBA currently uses to quantify how "optimal" our solution is. We use our evaluator to evaluate the schedule NBA currently uses, we then compare the result to that of our own generated schedule. Our solution gives a total revenue of 315889.5, whereas the actual 2016-2017 games give a value of 27150. There is a 16% increase in the revenue by using our generated schedule, but given that the weights in determining the total revenue are highly subjective, we would say the results of the two schedules are close.

The game we scheduled for Christmas night and the actual games played on Christmas night are very similar. This is because we assign a very high weight to this time slot. And there is a 39% overlap between the games we schedule for weekends and actual games played on weekends. This reasonably good agreement is probably due to the high value of weekend games and the algorithm matching high-value games with the high-value slot. However, the games we scheduled for weekdays and the actual games played on weekdays differ considerably. Comparing the revenues as a whole, there is not a huge difference in the objective considering that weights on different factors are subjective and we do not have enough information as the league officials do.

A possible path for further research is with variations in the problem. Currently, we are maximizing the overall revenue, conforming to the NBA's scheduling problem. However, it could also be interesting to consider an objective that tries to maximize the fairness of the game, either from the teams' or networks' perspective. These are all interesting and deep problems for further integer programming research.

6 Appendix:

Resources:

https://www.basketball-reference.com/leagues/NBA_2017_games.html

https://www.nbastuffer.com/analytics101/how-the-nba-schedule-is-made/

https://en.wikipedia.org/wiki/National_Basketball_Association

https://en.wikipedia.org/wiki/Hill_climbing

Code:

https://github.com/moqiand/21393-Project