GRP_123: Arranging a season for a team in NBA

YITING MA(20232544)

KEVIN JIANG(20215548)

PROUD SAWYER(TBA)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

October 29, 2021

Abstract

The NBA is the highest-level and most popular professional basketball league in the world. In general, each team needs to play 82 games during the regular season. This project aims to explain how to arrange a complete season for a team in NBA.

In our project, we select a team from Central zone, Eastern conference.

Propositions

D_i is the i^{th} day. if D_i is true, then there is a game in the i^{th} day.
$P_{(i,x)}$ is true if our team play on day i with team x.
B_i is true if there are back-to-back games starting at i^{th} day.
AS is true if the game is an All-Star game, which is three days long.
$C(x)$ is true if team x is in Central zone.
$ASE_6(x)$ is true if team x is within the six teams of Atlantic and Southeast.
$ASE_4(x)$ is true if team x is within the four teams of Atlantic and Southeast.
$W(x)$ is true if team x is in Western conference.

Constraints

If there are back-to-back games starting at the i^{th} day, then there will be no games on the $(i-1)^{th}$ day and $(i+2)^{th}$ day.

$$B_i \implies (\neg D_{i-1}) \land (\neg D_{i+2}) \land D_i \land (D_{i+1})$$

If All-Star games, then there will be no games for 3 days.

$$AS \implies (\neg D_{124} \land \neg D_{125} \land \neg D_{126})$$

If there is a game in the i^{th} day, then our team will play with team x.

$$D_i \implies P_{(x,i)}$$

If a team is in Central zone, then our team plays four matches with them.

$$C(x) \implies P_{(x,i_1)} \land P_{(x,i_2)} \land P_{(x,i_3)} \land P_{(x,i_4)}$$

If a team is within the six random team in Atlantic and Southeast, then our team plays four matches with them.

$$ASE_6(x) \implies P_{(x,i_1)} \land P_{(x,i_2)} \land P_{(x,i_3)} \land P_{(x,i_4)}$$

If a team is within the rest four random team in Atlantic and Southeast, then our team plays three matches with them.

$$ASE_4(x) \implies P_{(x,i_1)} \wedge P_{(x,i_2)} \wedge P_{(x,i_3)}$$

If a team is within the Western conference, then our team plays two matches with them.

$$ASE_4(x) \implies P_{(x,i_1)} \wedge P_{(x,i_2)}$$

If a D_i is true, then at most one of $P_{(i,x)}$ is true.

$$D_i \implies P_{(i,x_1)} \vee P_{(i,x_2)} \vee P_{(i,x_3)} \vee ... \vee P_{(i,x_{29})}$$

If our team is playing team x on day i, then it is not play any other game on that day.

$$P_{(i,x)} \implies \neg(P_{(i,x_1)} \vee P_{(i,x_2)} \vee ...P_{(i,x_{28})} \vee P_{(i,x_{29})})$$

If there is no game on day i, then the team we choose does not play any game on that day.

$$\neg D_i \implies \neg(P_{(i,x_1)} \wedge P_{(i,x_2)} \wedge ...P_{(i,x_{28})} \wedge P_{(i,x_{29})})$$

If there is a game on day i, then the team we choose plays a games with one team out of the total 29 teams.

$$D_i \implies (P_{(i,x_1)} \vee P_{(i,x_2)} \vee ...P(i,x_{28}) \vee P_{(i,x_{29})})$$

The first day of a season will always have a game.

$$D_1$$

If there is a game on day i and it is not a back to back game, then there is a game on day $i+2$ and no game on day $i+1$.

$$(D_i \wedge \neg B_i) \implies (\neg D_{(i+1)} \wedge D_{(i+2)})$$

There can not be a back to back game starting on the last day of the season.

$$\neg B_i \text{Where i is the last day of the season.}$$

Teams that are in Western Conference.

$$W_{(\text{Nuggets})} \wedge W_{(\text{Timberwolf})} \wedge W_{(\text{Thunder})} \wedge ...W_{(\text{Spurs})}$$

Model Exploration

We have explored our project in many different ways in terms of how to code our NBA schedule. As you will be able to see in our project code we have tested out many different ways to properly output and display and ordered NBA schedule. An example of our exploration is properly sorting the schedule, we tired many different ways to sort out the days.

When we started to implement the constraint into python, we realized the constraints we had are not enough to completely schedule the season. We only had a constraint on a back to back game, but not on how we will schedule a regular game.We realized that we do not have a way to initialize the season. It is like a recursion without a base case. We decided that we needed a template for the season. The template was meant to model a season with no back to back game. Which is if $Day(i)$ is ture, then $Day(i+1)$ is false, and $Day(i+2)$ is true. It means the basic schedule for a team is to play a game and rest a game. At this point, we could combine our back to back constraint with this base template to schedule a season. This model will tell us on which day there will be a game. At this point, we encountered a problem with counting the games that have been played. We could not create a correct constraint that will limit the total games to 82, or cause our team to play each other the correct amount of times. We thought it would be best to model how many times our team played with each other first, because if we played all the teams the correct times, we would end up having played 82 games. Unfortunately we were not successful at creating such a constraint. The constraint we created will cause the model to not be satisfiable. Right now, our group is focusing on how to set up the counting constraint.

First-Order Extension

Some extension we have thought about:

1. instead of doing a strict schedule of play 1 game break then play 1 games break we could make a team play at most a certain amount of games in a certain period of time example like 3 games in a 5 day period.

2. We could create a schedule for the entire NBA and have it make sense for the NBA to actually use.

3. Set up times for each team to play on each day. Example Bulls vs Knicks play at 7:30.

Feedback Request

Q1. For a NBA schedule every team needs to play exactly 82 games and they need to play each team a certain amount of times. We have tried everything we could think of and are still unable to create constraints that allow us to stratify this aspect of a NBA schedule. If you have any suggestions that would be a big break through for use.

Useful Notation

Feel free to copy/paste the symbols here and remove this section before submitting.

$$\wedge \quad \vee \quad \neg \quad \rightarrow \quad \forall \quad \exists$$