GRP_13: Arranging a season for a team in NBA

YITING MA(20232544)

KEVIN JIANG(20215548)

PROUD SAWYER(20199722)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

December 6, 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 the i^{th} game is played in the season with team x.
$BackToBack_i$ 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.

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

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.

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

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

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

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.

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

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.

$\neg BackToBack_i$ Where i is the last day of the season.

Teams that are in Western Conference.

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

Teams that are in Central Conference.

$$C_{(\text{Cavaliers})} \wedge C_{(\text{Pistons})} \wedge C_{(\text{Pacers})} \wedge ... \wedge C_{(\text{Bucks})}$$

Teams that are in ASE_6.

Similar with above.

Teams that are in ASE_4.

Similar with above.

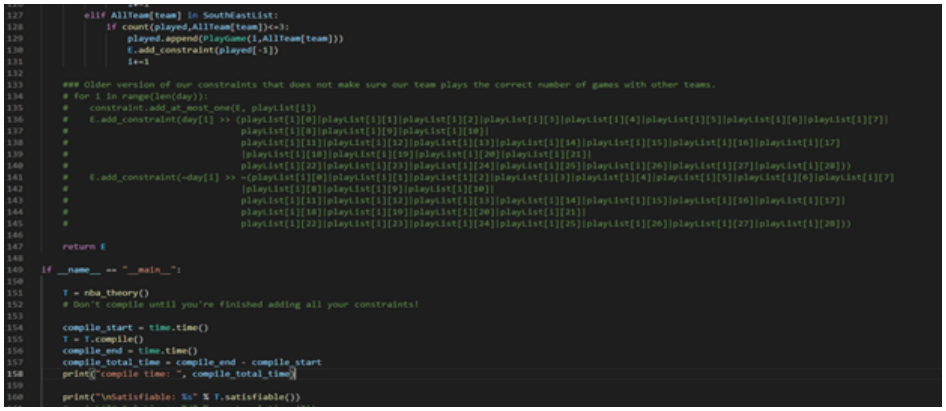
Model Exploration

We have explored our project in many 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 ways to properly output and display and ordered NBA schedule. An example of our exploration is properly sorting the schedule, we tired many ways to sort out the days.

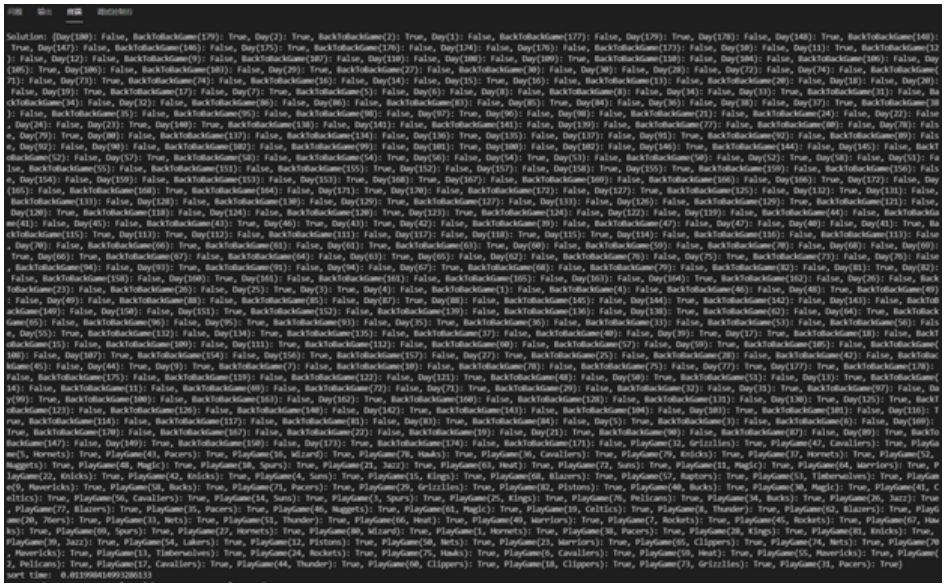
When we started to implement the constraint into python, we realized the constraints we had are not enough to completely schedule the season. We only had a constraint on a back-to-back game, but not on how we will schedule a regular game. We realized that we do not have a way to initialize the season. 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 true, 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 and it looked something like this.

Although it was only 10 days and a very confusing schedule we still got and output that made sense if broken down into its piece's. 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 number 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. We tried many different types of code such as:



We where able to create a chunks of code that properly counted to 82 games however these version of the code all ended up being terribly optimised, and result in memory error and extremely long wait times especially when trying to count 82 games. Thus we decided to use more python code in order to make our program run. Although we used more basic python coding, we still were able to include Bauhaus coding, so it was perfect for our project. After putting all our code together, we were able to generate a schedule that look like this that takes not time at all to generate.



Although it may look extremely confusing and disorganised this chuck of information was our perfect schedule now all we had to due was write code to organize the information so that I was more pleasing to the eye. Finally, after writing code that would properly sort the information our code generates the final working schedule that looks like this:

Game number:	Play with:	Play on day:
1	Nuggets	1
2	Bucks	3
3	Pacers	4
4	Jazz	6
5	Blazers	8
6	Grizzlies	9
7	Raptors	11
8	Magic	13
9	Bucks	15
10	Wizard	17
11	Thunder	18
12	Knicks	20
13	Blazers	22
14	Jazz	24
15	Grizzlies	26
16	Wizard	28
17	Grizzlies	30
18	Blazers	32
19	Cavaliers	34
20	Warriors	36
21	Pacers	38
22	Pacers	40
23	Suns	42
24	Clippers	44
25	Pistons	46
26	Nets	48
27	Pistons	49
28	76ers	51
29	Cavaliers	53
30	Lakers	55
31	Raptors	57
32	Pistons	59
33	Clippers	61
34	Pacers	62
35	Clippers	64
36	Nets	66
37	Cavaliers	67
38	Nets	69
39	Wizard	71
40	Lakers	73
41	Magic	74
42	Pelicans	76
43	Timberwolves	78

44	Timberwolves	80
45	Suns	82
46	Wizard	84
47	Spurs	86
48	Jazz	88
49	Nuggets	90
50	Heat	91
51	Cavaliers	93
52	Hornets	95
53	Pelicans	97
54	Lakers	99
55	Celtics	100
56	Hawks	102
57	Mavericks	104
58	Raptors	106
59	Celtics	108
60	Suns	109
61	Mavericks	111
62	Pistons	113
63	Raptors	114
64	Mavericks	116
65	Celtics	118
66	Knicks	120
67	Celtics	122
68	Nuggets	124
69	Heat	125
70	Cavaliers	127
71	Timberwolves	129
72	Magic	131
73	Kings	133
74	Pelicans	135
75	Wizard	137
76	Raptors	139
77	Pacers	141
78	Hornets	143
79	Hornets	144
80	Knicks	146
81	Hawks	148
82	Magic	150

Thus finishing our project.

Jape Proof

Proposition Symbols:

$BTBi$: True is there is a back to back game on day i.
Di: True is there is a game on day i.

Premises:
A back to back game on day i implies that there is a game on day i, $i+1$, but there is no game on day $i+2$.

$$BTB1 \rightarrow (D1 \wedge D2 \wedge \neg D3)$$
$$BTB2 \rightarrow (D2 \wedge D3)$$

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

$$(D1 \wedge \neg BTB1) \rightarrow (\neg D2 \wedge D3)$$
$$(D2 \wedge \neg BTB2) \rightarrow (\neg D3)$$

Proof:
The schedule can not contain three consecutive games.

$$BasePremise \vdash \neg(D1 \wedge D2 \wedge D3)$$

If there is a game on day1 , then there is a game on day 2 or 3.

$$BasePremise \vdash D1 \rightarrow (D2 \vee D3)$$

If there is a game on day 1 and 3, then there is not a back to back game on day 1 and 2

$$BasePremise \vdash (D1 \wedge D3) \rightarrow (\neg BTB1 \wedge \neg BTB2)$$

First-Order Extension

Proposition: $Day(i)$ True if there is a game on the i^{th} day of the season

$West(x)$ True if a team is in west conference

$Central(x)$ True if a team is in central conference

$ASE6(x)$ True if a team in Atlantic or southeast conference, and our team play 4 games with them

$ASE4(x)$ True if a team is in the Atlantic or southeast, and our team plays 3 games with them.

$BackToBack(i)$ true if there is a back to back game on the i^{th} day of the season

$PlayGame(i, x)$ true if the ith game of the season

$First(i)$, true if i is the first day of the season

$Last(i)$, true if i is the last day of the season

$SLast(i)$, true if i is the second last day of the season

Constraint: Constraint on $First(i)$, $Last(i)$, $SLast(i)$

For every possible day i, if i is the first day of the season, then it is not the second last day or the last day

$\forall i(First(i) \implies (\neg Last(i) \wedge \neg SLast(i)))$

For every possible day i, if i is the last day of the season, then it is not the second last, or the first day of the season

$\forall i(Last(i) \implies (\neg First(i) \wedge \neg SLast(i)))$

For every possible day i, if i is the second last day of the season , then it is not the last or the first day of the season

$\forall i(Last(i) \implies (\neg First(i) \wedge \neg Last(i)))$

Constraint on $BacktoBack(i)$:

For every possible day x in the season. If x is not the first day, seconde last day, and the last day of the season and there is a back to back game on that day, then there is a game on the x^{th} day, and $(x+1)^{th}$ day, and there is no game on x-1 and x+2 th day.

$\forall x((BackToBack(x) \wedge \neg First(x) \wedge \neg Last(x) \wedge \neg Slast(x)) \implies (Day(x) \wedge Day(x+1) \wedge \neg Day(x-1) \wedge \neg Day(x+2))$

For every possible day x in the season if x is the first day, and there is a back to back game on that day, then there is a game on day x, $x+1$, and there is no game on day $x+2$

$\forall x((BackToBack(x) \wedge First(x)) \implies (Day(x) \wedge Day(x+1) \wedge \neg Day(x+2))$

For every possible day x in the season. if x is the second last day of the season, and there is a back to back game on that day, then there is a game on day x, $x+1$, and there is no game on day $x-1$

$\forall x((BackToBack(x) \wedge Slast(x)) \implies (Day(x) \wedge Day(x+1) \wedge \neg Day(x-1))$

For every possible day x in the season, if x is the last day of the season, then there is no back to back game that starts on that day.

$\forall x(Last(x) \implies \neg BackToBack(x))$

Constraint on Day(i)

For every possible day x in the season. If the game on day x is not back to back , and there is a game on day x and it is not the second last, or last day of the season, then there is no game on the $(x+1)^th$ day, and there is a game on $(x+2)^{th}$ day

$\forall x((\neg BackToBack(x) \wedge Day(x) \wedge \neg Slast(x) \wedge \neg last(x)) \implies (\neg Day(x+1) \wedge Day(x+2)))$

For every possible day x in the season. If x is the second last day, and the game on day x is not back to back, and there is a game on day x, then there is no game on the $x+1$ day

$\forall x((\neg BackToBack(x) \wedge Day(x) \wedge Slast(x)) \implies \neg Day(x+1))$

Constraint on how many games to be played

For every team in the league, if it is in western conference, then our team plays two games with them.

$\forall x(West(x) \implies (\exists y_1 \exists y_2 .... \exists y_{82}(y_1 \neq y_2 \neq ... \neq y_{82})(PlayGame(y_1, x) \wedge PlayGame(y_2, x) \wedge \neg PlayGame(y_3, x) \wedge \neg PlayGame(y_4, x) \wedge \neg ..... \wedge \neg PlayGame(y_{82}, x))))$

For every team in the league, if it is in the central conference, then our team plays four games with them.

$\forall x(Central(x) \implies (\exists y_1 \exists y_2 .... \exists y_{82}(y_1 \neq y_2 \neq ... \neq y_{82})(PlayGame(y_1, x) \wedge PlayGame(y_2, x) \wedge PlayGame(y_3, x) \wedge PlayGame(y_4, x) \wedge \neg PlayGame(y_5, x) \wedge \neg PlayGame(y_6, x) \wedge \neg ..... \wedge \neg PlayGame(y_{82}, x))))$

Similar for team in ASE6 and ASE4

Let y and z be teams in the league and x be the x^{th} game in a season. If the x^{th} game that our team is playing in the season is against team y,

then our team is not playing team z as their x^{th} game of the season.

$$\forall x \forall y (PlayGame(x, y) \implies \forall z (Dif(z, y) \implies \neg PlayGame(x, z)))$$

There can only be 82 days that have a game on it

$$\exists i_1 \exists i_2 \exists i_3 .... \exists i_{180} (i_1 \neq i_2 \neq i_3 ... \neq i_{180}) (Day(i_1) \wedge Day(i_2) \wedge Day(i_3) ... \wedge Day(i_{82}) \wedge \neg Day(i_{83}) \wedge \neg Day(i_{84}) \wedge \neg .... \wedge \neg Day(i_{180}))$$