

CS/ECE/ISyE 524 — Introduction to Optimization — Spring 2020

NBA Fantasy Team Optimizer

Table of Contents

1. [Introduction](#)
2. [Mathematical Model](#)
3. [Solution](#)
4. [Results and Discussion](#)
5. [Conclusion](#)

1. Introduction

In the 1980's, a New York City sportswriter named Daniel Okrent created fantasy sports with the idea of allowing sports fans to experience a side of sports that they had never experienced. Fantasy sports are internet simulation-based games where players assemble virtual teams of actual professional players. These professional players are ranked by their statistical in-game performance and are then selected by users who place them on their respective teams. These teams then compete against one another based on the performance of team's players in actual games. In a standard fantasy league, winning games results in points that a team gets to keep. At the end of the fantasy season, the team with the most points wins the league (the championship). In fantasy basketball, a **fantasy score** is a metric that considers the fundamental statistics (games played, field goal %, three-point%, points per game, rebounds, turnovers, and more) in a basketball game as a number.

R#	PLAYER	POS	TEAM	GP	MPG	FG%	FT%	3PM	PTS	TREB	AST	STL	BLK	TO	TOTAL
1	James Harden	PG,SG	HOU	61	36.7	0.44 (9.9/22.7)	0.86 (15.1/17.6)	4.4	34.4	6.3	1.4	1.7	0.9	4.4	15.84

In the above image, **Total** corresponds to the player's fantasy score.

For the past couple of years, fantasy basketball has been growing with increasing popularity. This emerging increase has resulted in the demand of fantasy players to find a way to increase their chances of success with their fantasy team. From simulation software, to predictive analytics, many different tools/methods currently exist to aid fantasy users on which players to select onto their teams, each with their own pros and cons. To address this demand, our group attempted to create the ideal tool that fantasy players can use to build their perfect team.

For this project, we have built an optimization model that takes all 450 current players in the NBA and builds the best fantasy team possible in order to guarantee success in a fantasy basketball league. This model considers each player's fantasy score, position, salary, team, and more to find whether or not a respective player is the right fit. Additionally, the model also considers which players are best suited to come off the bench as "backup", when certain

players are injured or are not playing.

We collected our data, including in-game statistics, player salary, position, and more on all NBA players in the 2019-2020 season from [Hashtag Basketball](https://hashtagbasketball.com/fantasy-basketball-rankings) (<https://hashtagbasketball.com/fantasy-basketball-rankings>) and [Basketball-Reference](https://www.basketball-reference.com/) (<https://www.basketball-reference.com/>). Both websites refresh and update their data logs daily (while the season is ongoing), ensuring data accuracy and reliability. In order to collect this data, we copy the appropriate stat charts and place them into an Excel spreadsheet.

2. Mathematical model

Since we are trying to create a fantasy team using players with high fantasy scores, we want to **maximize the team's average fantasy score**, subject to the applicable constraints. There is one set of primary decision variables (**selected**) and 3 major constraints:

(1) The number of players on a team

In basketball, a team must have 5 of its players on the floor at any given time. Our optimization model will consider the number of open spots on the team, and will select players to fit those positions. Additionally, the model will also choose *bench players* that provide backup for the starting 5.

(2) A salary constraint considering the NBA's current salary cap

As NBA players are known for their lucrative salaries, teams must decide what players they choose based on what they can afford. In the NBA, a **salary cap** is *the maximum range of money that a team is allowed to spend on their players*. In the 2019-2020 season, the NBA salary cap falls between **109 Million to 132 Million US Dollars**. Our optimization model will choose players, considering their respective salaries, to build the optimal fantasy team while falling within the recommended team salary range.

(3) A position constraint requiring that all 5 positions on the court are filled by appropriate players.

In basketball, there are 5 positions on the court: Point Guard (PG), Shooting Guard (SG), Small Forward/Forward (SF/F), Power Forward (PF), and Center (C). The optimization model will pick players that can play these positions, as well as choose backup players in the case where our starters are injured or are not playing in a game.

All of these constraints will be applied to consider every single player in the NBA in order to create the most ideal fantasy team. The problem could be described as the following model:

$$\begin{aligned}
 &\underset{\text{selected}}{\text{maximize}} && \sum_{n=1}^{12} \text{selected} * \text{rank}_i \\
 &\text{subject to:} && \sum_s \text{selected}_i == 12 \\
 &&& \$109M \leq (\sum \text{selected}_i * \text{salary}_i) \leq \$132M \\
 &&& \sum \text{selected}_i * \text{pos}_{i,j} \geq, \text{pos_requirement}_j
 \end{aligned}$$

where:

P is the number of players in our dataset, which is about 450.

selected is a P x 1 matrix representing whether a player is chosen to be on the team; assigned a **1** if they are selected, **0** if not.

rank is a P x 1 matrix representing the overall fantasy score for each player, with the maximum rank being 100.

- **rank_i ≤ 100**

salary is a P x 1 matrix that contains the respective salaries for each player in *millions of US Dollars*.

- \$109Million & \$132 Million are the respective lower and upper bounds for the NBA Salary Cap for the 2019-2020 season.

position is a P x 5 matrix that corresponds to what position(s) a player can be assigned to.

- **position[i j]** is assigned a **1** if player *i* can play position *j*
- assigned a **0** if not.

3. Solution

Part A - Fantasy Score Optimizer

Below, we import the dataset containing information on every active player in the 2019-2020 NBA season. After importing the data, we immediately create new matrices that consider each player's respective position, salary, and fantasy score.

```
In [1]: ▶ using CSV
nba = CSV.read("nba.csv", header = 1, types=Dict(:Salary => Int));
position = nba[!, [:C, :PF, :SF, :SG, :PG]];
Salary = nba[:, :Salary];
```

12 Player Optimizer

```

In [2]: ▶ using JuMP, Gurobi
m = Model(with_optimizer(Gurobi.Optimizer, OutputFlag = false))
@variable(m, selected[1:444], Bin)

# Select 12 players
@constraint(m, sum(selected[i] for i in 1:444) == 12)

# Salary Constraint
@constraint(m, 109000000 .<= sum(selected[i] * Salary[i] for i in 1:444) .<= 132000000)

# More than 6 guards in a team
@constraint(m, sum(selected[i] * (position[i,:SG]+ position[i,:PG]) for i in 1:444) .>= 6)

# More than 4 forwards
@constraint(m, sum(selected[i] * (position[i,:SF]+ position[i,:PF]) for i in 1:444) .>= 4)

# At least one center
@constraint(m, sum(selected[i] * (position[i,:C]) for i in 1:444) .>= 1)

# Maximize score
@objective(m, Max, sum(selected[i]*Score[i] for i in 1:444))
;

# Optimize
optimize!(m)
println("termination status: ", termination_status(m))
println()

# Assign players to team and print names
y = JuMP.value.(selected)
# Pick up if the esimated value >0.5
label = findall(x->x>0, y)

println("NBA FANTASY TEAM WITH 12 PLAYERS")
println()
println("Players: ", nba[label, :Player])
println()
println("Total Salary: \$", sum(Salary[label, :]))

```

Academic license - for non-commercial use only

Academic license - for non-commercial use only

termination status: OPTIMAL

NBA FANTASY TEAM WITH 12 PLAYERS

Players: ["James Harden", "Anthony Davis", "Trae Young", "John Collins?", "Luka Doncic", "Brandon Ingram", "Jayson Tatum", "Deandre Ayton?", "Jonathan Isaac?", "Pascal Siakam", "Ben Simmons?", "Bam Adebayo"]

Total Salary: \$125920633

Using the **12-Player Fantasy Score Optimizer**, the ideal fantasy team would consist of:

James Harden (PG), Anthony Davis (PF), Trae Young (PG), John Collins (PF), Luka Doncic (PG), Brandon Ingram (SF), Jayson Tatum (SF), Deandre Ayton (C), Jonathan Isaac (PF), Pascal Siakam (PF), Ben Simmons (PG), and Bam Adebayo (PF).

The team salary for this 12 player roster would be **\$125,920,633** - which perfectly falls within the NBA salary cap.

8 Player Optimizer

When doing our research on fantasy basketball, our group noticed that most commonly, a majority of fantasy basketball teams consisted of either **12** or **8** players. In order to make our model useful to fantasy users who desire to create teams with exactly 8 players, we created our **8-Player Fantasy Score Optimizer** below.

```
In [3]: ▶ using JuMP, Gurobi
m = Model(with_optimizer(Gurobi.Optimizer, OutputFlag = false))

@variable(m, selected[1:444], Bin)

#PICK UP 8 PLAYERS THIS TIME instead of 12
@constraint(m, sum(selected[i] for i in 1:444) == 8)

# Salary Constraint
@constraint(m, 109000000 .<= sum(selected[i] * Salary[i] for i in 1:444) .<= 132000000)

# More than 3 guards on the team
@constraint(m, sum(selected[i] * (position[i,:SG]+ position[i,:PG]) for i in 1:444) .>= 3)

# More than 4 forwards on the team
@constraint(m, sum(selected[i] * (position[i,:SF]+ position[i,:PF]) for i in 1:444) .>= 4)

# At least one center
@constraint(m, sum(selected[i] * (position[i,:C]) for i in 1:444) .>= 1)

# Maximize the score
@objective(m, Max, sum(selected[i]*Score[i] for i in 1:444))
;
# Optimize

optimize!(m)
println()
println("Termination Status: ", termination_status(m))
println()

# Assign players and print player names for the 8-player team, as well as the salary
y = JuMP.value.(selected)
label = findall(x->x>0, y) # pick up if the esimated value >0.5

println("NBA FANTASY TEAM WITH 8 PLAYERS")
println()
println("Players: ", nba[label, :Player])
println()
println("Total Salary: \$", sum(Salary[label, :]))
```

Academic license - for non-commercial use only
Academic license - for non-commercial use only

Using the **8-Player Fantasy Score Optimizer**, the ideal fantasy team would consist of: *James Harden (PG), Anthony Davis (PF), Damian Lillard (PG), Trae Young (PG), John Collins (PF), Luka Doncic (PG), Brandon Ingram (SF), and Pascal Siakam (PF) or Jayson Tatum (SF).*

The salary for this 8 player roster would be **\$126,433,745** - and it falls within the recommended salary cap.

Part B - Advanced Statistics Optimizer

Now that we have successfully found a way to use a player's **fantasy score** to create the best fantasy team for the 2019-2020 NBA season, our group continued to look for more ways to take our model a step further.

In the realm of fantasy basketball, there are numerous users that are critics of the fantasy score, as they claim that it is a statistic that condenses a player's performance into a single number -- which is a somewhat flawed and arbitrary measure of a player's gameday performance.

In order to address those critics, our group introduces Part B of our model, the **Advanced Statistics Optimizer**. Within this model, we created a new dataset that builds on the existing one above, adding the following statistics:

- Assists
- Attempted Field Goals
- Attempted Free Throws
- Attempted Three-Point Field Goals
- Blocks
- Defensive Rebounds
- Offensive Rebounds
- Games Played
- Games Started
- Made Field Goals
- Made Free Throws
- Made Three Point Field Goals
- Minutes Played
- Points
- Fouls

To obtain and assign weights associated to each of the statistics, we use a **Linear Regression Model** to determine the weights of each feature.

Below, we decide the weights of each feature. Using regression as an indicator variable, which indicates whether a team won the NBA Championship that season, by using the team's advanced statistics that season. *In order to run the below code block, you will need to install the **MLDataUtils** package.*

Once the data is loaded successfully, we split it into a training and test set using a ratio of 80% for the training of the model with 20% withheld for the evaluation of the generated model.

```
In [4]: ▶ using Pkg
# Pkg.add("MLDataUtils")
using CSV, MLDataUtils, JuMP, Statistics

# team_features: 1990-2019 team average stats
nba = CSV.read("team_features.csv", header = 1);
y = nba[:champion];
X = nba[:,value_over_replacement_player,:win_shares, :defensive_win_shares,
        :win_shares_per_48_minutes, :offensive_win_shares, :age,
        :defensive_box_plus_minus]]

# Split training (80%) and testing dataset (20%)
(X_train, y_train), (X_test, y_test) = splitobs((X, y); at = 0.8);

# Convert to matrices
X = convert(Matrix, X_train[:,:])
y = convert(Matrix, y_train[:,])

# Compare sizes of training and testing datasets
println("training size: ", size(X_train));

└─ Warning: `getindex(df::DataFrame, col_ind::ColumnIndex)` is deprecated, use
`df[!, col_ind]` instead.
| caller = top-level scope at In[4]:7
└─ @ Core In[4]:7
└─ Warning: `getindex(df::DataFrame, col_inds::Union{AbstractVector, Regex, No
t})` is deprecated, use `df[:, col_inds]` instead.
| caller = top-level scope at In[4]:8
└─ @ Core In[4]:8

training size: (698, 7)
testing size: (175, 7)
```

More on data normalization can be found [here \(https://towardsdatascience.com/julia-for-data-science-regularized-logistic-regression-667857a7f0ce\)](https://towardsdatascience.com/julia-for-data-science-regularized-logistic-regression-667857a7f0ce).

```
In [5]: ▶ """
scale_features(X)
This function attempts to standardise the design matrix (X) user pass.
The input data X is standardised and returned along with other learned matrices.
A tuple with 3 elements is returned representing (Standardised data, mean, std d
"""

function scale_features(X)
    μ = mean(X, dims=1)
    σ = std(X, dims=1)

    X_norm = (X .- μ) ./ σ

    return (X_norm, μ, σ);
end

# Normalise the testing design matrix
```

```

"""
transform_features(X,  $\mu$ ,  $\sigma$ )
This function uses the mean and standard deviation values users pass to normalise
"""
function transform_features(X,  $\mu$ ,  $\sigma$ )
    X_norm = (X .-  $\mu$ ) ./  $\sigma$ 
    return X_norm;
end

# Scale training features and get artificats for future use
X,  $\mu$ ,  $\sigma$  = scale_features(X);

```

Linear Regression Weights:

In [6]: ▶

```

Out[6]: 7x1 Array{Float64,2}:
 0.094905651979068
-0.6560783344444937
 0.35135673872718304
 0.009231920939974447
 0.3667155913617212
 0.012967247111781188
-0.008366665259845923

```

Generate the index of each players based on their respective stats and the weights calculated above.

In [7]: ▶

```

# combined: Stats of All NBA players in 2019-2020 season
nba = CSV.read("combined.csv", header = 1, types=Dict{ :Salary => Int});
position = nba[!, [:C, :PF, :SF, :SG, :PG]];
Salary = nba[:, :Salary];

x = nba[!, [:value_over_replacement_player, :win_shares, :defensive_win_shares,
            :win_shares_per_48_minutes, :offensive_win_shares, :age,
            :defensive_box_plus_minus]]

x = convert(Matrix,x);
x = transform_features(x,  $\mu$ ,  $\sigma$ )

```

12-Player Advanced Statistics Optimizer

Generate a 12-player Fantasy Team by maximizing the index under the constraints of position, salary, and team budget, like we did in *Part A*.

In [8]: ▶

```

using JuMP, Gurobi
n = size(nba)[1]
m = Model(with_optimizer(Gurobi.Optimizer, OutputFlag = 0))
@variable(m, selected[1:n], Bin)

#pick up 12 players
@constraint(m, sum(selected[i] for i in 1:n) == 12)

# Salary Constraint
@constraint(m, 109000000 .<= sum(selected[i] * Salary[i] for i in 1:n) .<= 13200

```



```

# More than 6 guards in a team
@constraint(m, sum(selected[i] * (position[i,:SG]+ position[i,:PG])) for i in 1:n

# More than 4 fronts
@constraint(m, sum(selected[i] * (position[i,:SF]+ position[i,:PF])) for i in 1:n

# At least one center
@constraint(m, sum(selected[i] * (position[i,:C])) for i in 1:n) .>= 1)

# Maximize score
@objective(m, Max, sum(selected[i]*Index[i] for i in 1:n))

# Optimize
optimize!(m)
println("termination status: ", termination_status(m))

# Assign players and print player names for the 12-player team
y = JuMP.value.(selected)
label = findall(x->x>0, y)
#nba[label, :name]

println()
println("NBA FANTASY TEAM USING 12-PLAYER ADVANCED STATISTICS OPTIMIZER")
println()
# Print team - players names
println("Players: ", nba[label, :name])
println()
# Print new team - total salary

Academic license - for non-commercial use only
Academic license - for non-commercial use only
termination status: OPTIMAL

```

NBA FANTASY TEAM USING 12-PLAYER ADVANCED STATISTICS OPTIMIZER

Players: ["Luka Doncic", "Jayson Tatum", "Trae Young", "Bam Adebayo", "Donovan Mitchell", "Domantas Sabonis", "Nikola Jokic", "Giannis Antetokounmpo", "Pascal Siakam", "Buddy Hield", "George Hill", "LeBron James"]


Total Salary: \$130471952

Using the **12-Player Advanced Statistics Optimizer**, the ideal fantasy team would consist of: *Bam Adebayo (PF), Giannis Antetokounmpo (PF/SF), Luka Doncic (PG), Buddy Hield (SG), George Hill (PG), LeBron James (PF/SF), Nikola Jokic (PF/C), Donovan Mitchell (PG), Domantas Sabonis (PF), Pascal Siakam (PF), Jayson Tatum (SF), and Trae Young (PG).*

The salary for this 12-player roster would be **\$130,471,952**, and it falls within the NBA Salary Cap.

8-Player Advanced Statistics Optimizer

Generate a 8-player Fantasy Team by maximizing the index under the constraints of position, salary, and team budget, like we did in *Part A*.

In [9]:  `using JuMP, Gurobi
m = Model(with_optimizer(Gurobi.Optimizer,OutputFlag = false))
@variable(m, selected[1:444], Bin)`

```

    #pick up 8 players
    @constraint(m, sum(selected[i] for i in 1:444) == 8)

    # Salary Constraint
    @constraint(m, 109000000 .<= sum(selected[i] * Salary[i] for i in 1:444) .<= 132000000)

    # More than 6 guards in a team
    @constraint(m, sum(selected[i] * (position[i,:SG]+ position[i,:PG]) for i in 1:444) .<= 6)

    # More than 4 fronts
    @constraint(m, sum(selected[i] * (position[i,:SF]+ position[i,:PF]) for i in 1:444) .<= 4)

    # At least one center
    @constraint(m, sum(selected[i] * (position[i,:C]) for i in 1:444) .>= 1)

    # Maximize score
    @objective(m, Max, sum(selected[i]*Index[i] for i in 1:444))
    optimize!(m)
    println("termination status: ", termination_status(m))

    y = JuMP.value.(selected)
    label = findall(x->x>0, y)
    #nba[label, :name], sum(Salary[label, :])

    println()
    println("NBA FANTASY TEAM USING 8-PLAYER ADVANCED STATISTICS OPTIMIZER")
    println()
    # Print team - players names
    println("Players: ", nba[label, :name])
    println()
    # Print new team - total salary

```

Academic license - for non-commercial use only

Academic license - for non-commercial use only

termination status: OPTIMAL

NBA FANTASY TEAM USING 8-PLAYER ADVANCED STATISTICS OPTIMIZER

Players: ["Luka Doncic", "Jayson Tatum", "Bam Adebayo", "Nikola Jokic", "Giannis Antetokounmpo", "Pascal Siakam", "Anthony Davis", "Damian Lillard"]

Total Salary: \$130630911

Using the **8-Player Advanced Statistics Optimizer**, the ideal fantasy team would consist of: *Bam Adebayo (PF)*, *Giannis Antetokounmpo (PF/SF)*, *Luka Doncic (PG)*, *James Harden (PG)*, *LeBron James (PF/SF)*, *Ben Simmons (PG)*, *Jayson Tatum (SF)*, and *Pascal Siakam (PF) or Damian Lillard (PG)*.

The salary for this 8-player roster would be **\$130,512,764**, and it aligns with the NBA Salary Cap.

4. Results and discussion

This project implements two optimization model. The first model is a straightforward Linear Program where we have applied the constraints of (1) number of players (2) salary cap (3) (various) positions of players. The second model uses Linear Regression Model with

additional statistical data about top 10 closely related features.

Part A: Fantasy Score Optimizer

This optimizer takes into account the players' salaries, their fantasy score and their playing positions. A fantasy score is a metric that considers the fundamental statistics (games played, field goal %, three-point%, points per game, rebounds, turnovers, and more) in a basketball game

12-Player Fantasy Team

Players: James Harden (PG), Anthony Davis (PF), Trae Young (PG), John Collins (PF), Luka Doncic (PG), Brandon Ingram (SF), Jayson Tatum (SF), Deandre Ayton (C), Jonathan Isaac (PF), Pascal Siakam (PF), Ben Simmons (PG), and Bam Adebayo (PF).

Salary: The team salary for this 12 player roster would be **\$125,920,633** - which perfectly falls within the NBA salary cap.

8-Player Fantasy Team

Players: James Harden (PG), Anthony Davis (PF), Karl-Anthony Towns (PF), Trae Young (PG), John Collins (PF), Luka Doncic (PG), Brandon Ingram (SF), and Pascal Siakam (PF).

Salary: The salary for this 8 player roster would be **\$126,433,745** - and it falls within the recommended salary cap.

Part B: Advanced Statistics Optimizer

This optimizer uses a new dataset that builds on the existing one above, using additional statistics. We first figure out the top 10 highly correlated features. Considering the below listed features, our objective remains the same as before - to maximize the score.

- Assists
- Attempted Field Goals
- Attempted Free Throws
- Attempted Three-Point Field Goals
- Blocks
- Defensive Rebounds
- Offensive Rebounds
- Games Played
- Games Started
- Made Field Goals
- Made Free Throws
- Made Three Point Field Goals
- Minutes Played
- Points
- Fouls

12-Player Fantasy Team

Players: Bam Adebayo (PF), Giannis Antetokounmpo (PF/SF), Luka Doncic (PG), Buddy Hield (SF), George Hill (PG), LeBron James (PF/SF), Nikola Jokic (PF), Donovan Mitchell (PG), Domantas Sabonis (PF), Pascal Siakam (PF), Jayson Tatum (SF), and Trae Young

(PG).

Salary: The salary for this 12-player roster would be **\$130,471,952** and it falls within the NBA Salary Cap.

8-Player Fantasy Team

Players: Players in 12-Player Fantasy Team: Bam Adebayo (PF), Giannis Antetokounmpo (PF/SF), Luka Doncic (PG), James Harden (PG), LeBron James (PF/SF), Pascal Siakam (PF), Ben Simmons (PG), and Jayson Tatum (PG).

Salary: The salary for this 8-player roster would be **\$130,512,764** and it is within the NBA Salary Cap.

Discussion: NBA 2019-2020 Season

The current pandemic affects the finances of NBA season drastically. The collective bargaining agreement between the league and the players guarantees players a certain percentage of the league's basketball-related income (BRI) every year. That can range from 49-51 percent and most of that figure goes into the salary cap. However, games played without fans in the stands would cost the league a fortune not only in ticket sales, but in food, beverage, parking and merchandise sales. Television ratings might soar in a socially-distanced society, but with rights locked up years in advance, the NBA itself won't see the benefits. Its partners at ESPN and Turner News Network (TNT) would. What if games are called off entirely? That's money that the league has to return. In reality, the burden of loss will be shared by the league and the players. This would largely affect the salary caps for coming years.

One possibility is that rather than drastically lowering the cap all at once, the NBA could build the losses from this season into the cap for the next several years. In some form or another,

5. Conclusion

The overall purpose of this project was to solve a *real-life* optimization problem using the fundamentals that we have learned in class this semester. In our group's project, the **NBA Fantasy Optimizer**, we successfully created an optimization model that takes in a dataset of NBA players, their fantasy score, salary, position, and more, to create the optimal fantasy team that would yield success in the most competitive fantasy leagues. We created this model as a tool to serve fantasy basketball users worldwide, and are confident with its functionality.

To take things a step further, we also created another version of our model to consider all of the in-game statistics that are recorded on behalf of each player in the NBA. The new model uses linear regression to assign weights to each of the statistics, then creates the optimal fantasy team. When looking at the fantasy teams that are created from running our model(s), one would recognize the several All-Star/MVP caliber players that are listed, displaying the depth and strength of the fantasy team that our model(s) output. As a group, we are pleased with the quality of results that our optimization model gives us, and would love to see others input their own NBA datasets into the model to enhance their fantasy teams.

Going forward, our group came up with an idea that could be an interesting add-on to our existing model. Currently, the optimization model only has constraints based on position, fantasy score, advanced statistics, and salary. But what if a fantasy user wanted to apply

another constraint of their choice to the model? This new idea would allow them to do so.

*For example, if a fantasy user wanted to create the best possible NBA fantasy team using only players younger than 25 years old, he/she could use a new constraint called **AGE** which would only consider players that are at or below a selected age. Additionally, if a user only wanted to select players for certain teams, he/she could use a new constraint called **TEAM**, which would be a dictionary containing a list of desired teams to select players from.*

If this follow-up idea was implemented, fantasy users would be able to utilize new constraints like those above to enhance their experience with the model.

6. References / Citations

- [1] [Basketball Reference \(https://www.basketball-reference.com/\)](https://www.basketball-reference.com/)
- [2] [Hashtag Basketball \(https://hashtagbasketball.com/fantasy-basketball-rankings\)](https://hashtagbasketball.com/fantasy-basketball-rankings)
- [3] [FantasyPros \(https://www.fantasypros.com/nba/rankings/overall.php\)](https://www.fantasypros.com/nba/rankings/overall.php)
- [4] [Optimizing Fantasy Basketball - Berkeley Statistics \(https://www.stat.berkeley.edu/~aldous/157/Old_Projects/lu-zhang.pdf\)](https://www.stat.berkeley.edu/~aldous/157/Old_Projects/lu-zhang.pdf)
- [5] [ESPN - How to use the NBA schedule to win in fantasy hoops \(https://www.espn.com/fantasy/basketball/story/_/id/18018706/fantasy-basketball-how-use-nba-schedule-win-fantasy-hoops\)](https://www.espn.com/fantasy/basketball/story/_/id/18018706/fantasy-basketball-how-use-nba-schedule-win-fantasy-hoops)
- [6] [Drafting a Fantasy Basketball Team With Help From Statistics and a Knapsack \(https://medium.com/fun-with-data-and-stats/drafting-a-fantasy-basketball-team-c94967464908\)](https://medium.com/fun-with-data-and-stats/drafting-a-fantasy-basketball-team-c94967464908)