Daily Fantasy Sports Line-up Generation

Phuc Xuan Nguyen, Silu Yang, Nitin Agarwal, Kaining Yang

December 11, 2015

Abstract

Daily Fantasy Sports has recently become popular. Sites like DraftKings and Fanduel post daily prizes ranging from $1 to $1 million. In this project, we attempt to produce a line-up based on the projected fantasy points.

1 Problem Description

In recent years, Daily Fantasy Sports (DFS) has gained much popularity in America. A user can enter daily and weekly fantasy sports-related contests and win money. A user participating in a daily game is given a salary cap at $50000 and a set of players with different statistics. The goal is to produce the optimal line-up within the salary cap. This users then submit the lineup to different contests and the line-up with the best total of fantasy points for that particular day win. Figure 1 shows the daily flow of a DFS content.

In this project, we aim to automatically produce a DFS line-up for NBA fantasy basketball. Based on the existing history, we project the fantasy points for all the players using different methods. Using these projected fantasy points, we solve different Integer Programming Problems using different optimization methods.

2 Related Works

This section describes systems and methods designed to predict various aspects of the fantasy sports leagues. Morris et al. (2005) describe a system for predicting game play performance They provide a tool with good user interface and large data set. Ware & Webb (2006) introduce a tool for evaluating whether a fantasy team owner should add a player to the starting line up. It uses data from previous season and make fantasy points projection. Krieger & Wirtschafter (2008) describe a prediction engine that can generate relationships among the criteria associated with players and provide projections based on the established relationships. Kasten (2009) allow users to identify parameters for evaluating. The system learns and use the weighting factors

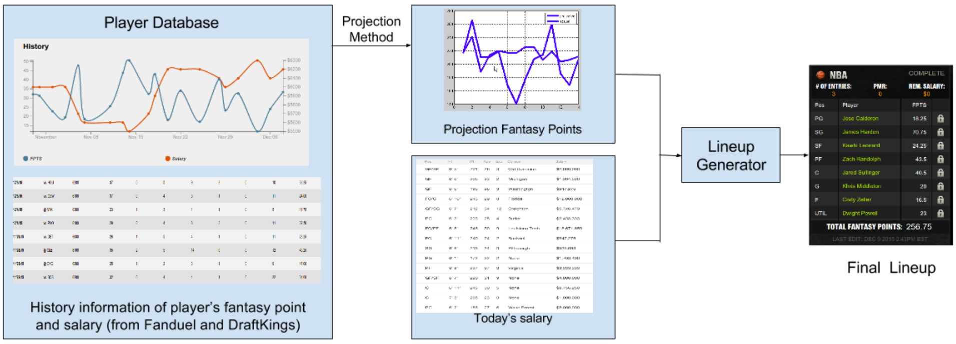

Figure 1: The daily flow for NBA fantasy game. The player database, consisting of statistics, salary, and fantasy logs, is available to the user. Using these information, we project the fantasy points for the current day. With the projected fantasy points and the salaries for the current day, we propose the optimal line-up.

in conjunction with historical data to determine model parameters. These parameters are lated used to rank players and help users to make lineup selection.

3 Contributions

In this section, we discuss the contributions of the team members. Phuc and Kaining are responsible for collecting and preprocessing the data. Phuc is also responsible for experimenting with different projection methods and their parameters. We experiment with 4 different possible line-up generators. Phuc is responsible for two baseline generators. Phuc and Nitin works on an optimization for integer programming problem with ad-hoc positions modifications. Silu codes up the Simulated Annealing algorithm to solve the optimization where the positions are factored into the problem formulation. Everyone participates in writing the report.

4 Collection process

We obtain 35 days of fantasy records for both Fanduel and DraftKings via Rotoguru.com. We use the first 20 days for learning projections and other hyper-parameters. The last 15 days are used for testing our optimization methods. For each day, we collect the positions, fantasy points, salary, the minutes played and several other statistics for all the players playing that day.

	Average	Last	Regression
All Error	6.62	8.58	6.74
Top 50 Errors	8.24	10.72	8.29

Table 1: Fantasy points errors for different projection methods.

5 Projection Methods

All of our line-up generation strategies rely on the projected fantasy points to make decisions for line up. We hypothesize that better projections would allow better optimization in the later part. In this section, we discuss several methods for projecting the fantasy points.

Average: We compute a player's projected fantasy point by averaging all his fantasy points history.

Last: The players' projected fantasy point is set to be his fantasy point from his last played game.

Regression We use linear regression to predict the projected fantasy point. With W as the window size, the features includes the fantasy points and minutes from W-day, the averaged fantasy points this seasons, and a bias. Figure 2 shows the average errors for different value of W. Small window size of 2-3 previous game seem to yield the best results.

5.1 Projection Comparisons

In this section, we compare the effectiveness of each projection methods. We use two metrics for comparing the projection errors: the mean errors for all the players, and the means errors for the top 50 players. The error is calculated as the L1-distance between the projected fantasy points and the ground-truth fantasy points. The reasoning for the metric is that most line-up generators are designed to select the best players (with the highest projected errors) for that day. Table 1 shows the errors for different project methods. Averaging the fantasy point history gives the best approximation for the projected scores. We use the FanDuel dataset in this experiment.

6 Line-up Generators

In this section, we described 4 different line-up generation methods. First, we describe two simple baselines: (1) choosing using averaged salaries and (2) initiating with the best three players. We, then, show two different formulations of the problems and their respected solvers.

Rules of the game Given a salary cap at $50000, a valid line-up must have the total salary less than cap and contains all of following 8 positions, Point Guard (PG), Shooting Guard (SG), Small Forward (SF), Power Forward (PF), Center (C), Guard (G), Forward (F), Utilities (UTIL). The

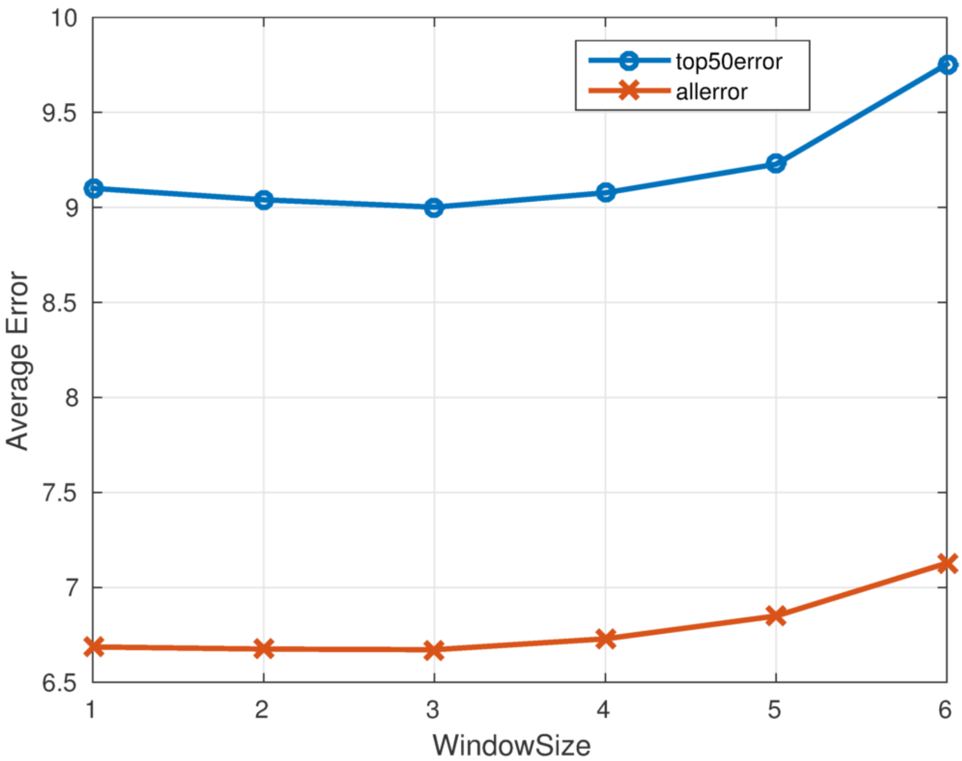

Figure 2: Regression Error over different window size.

Guard (G), Forward (F) and Utilities (UTIL) are flexible spots and can be filled by any PG or SG, any SF or PF and any players respectively.

6.1 Baseline Strategies

We include the following baselines for line-up generation.

Average Salaries We select each position using an average salary cap, salary cap divided by the number of positions. For each position, we greedily pick a player that fits the position, has smaller or equal salary compare to the averaged salary cap, and has the best projected fantasy points.

Big Three Many championship success in NBA basketball has been attributed to a phenomenon called the Big-3. Most championship contender teams often have 3 superstars. We designed a baseline following this concept. In this baseline, we select top 10 superstars, i.e. players with the highest salaries. Out of these 10 superstars, we pick 3 with the best values, measured by projected fantasy points per $1000 salaries. For the remaining positions, we use the averaged salary strategy mentioned in the previous paragraph.

6.2 Integer Linear Programming with Ad-Hoc Modifications

In this section, we describe our first problem formulation for line-up generation.

Mathematically, the problem is formulated as

$$\begin{aligned} \min_x \quad & f(x) = -\rho^T x \\ \text{subject to} \quad & S^T x \leq C, \\ & 1^T x = 8, \\ & x \in \{0, 1\}, \end{aligned}$$

with C is the salary cap, S is the salary for player i, and ρ is the daily projected fantasy points. This formulation satisfies the salary constraints, but not the position constraints. The solution could output invalid lineup.

We correct the solution via a simple heuristic. We sort the selected players based on their projected values and add, if allowed, players in this order to the line-up. For each missing position, we replace one of the remaining players, with a new player with the same or lower salary. If there are more than one new player that satisfy this conditions, we select the ones with the highest projected fantasy points.

We use the mixed-integer linear programming solver in Matlab, intlinprog, to solve for the solutions. The solver is easy to use and converges quickly.

6.3 Integer Linear Programming with Simulated Annealing

Instead of ad-hoc modifications to the line-up to satisfy the position constraints, we add the constraints to our problem formulation.

Using similar notations in previous section, the problem is defined as

$$\begin{aligned} \min_x \quad & f(x) = -\rho^T x \\ \text{subject to} \quad & S^T x \leq C \\ & 1^T x = 8 \\ & N_p \geq 1, \quad \forall p \in P \\ & N_{SF} + N_{PF} \geq 3 \\ & N_{SG} + N_{PG} \geq 3 \end{aligned}$$

with N_p denoting the number of players filled in position p, and $P = \{SF, PF, SG, PG, C\}$.

Simulated Annealing We solve the above formulation using Simulated Annealing (SA). Table 2 shows the performance of different values of the minimum temperature, T_{min}, and cooling off rate, γ. We see that a slower cooling rate and smaller minimum temperature leads better solutions. We

Cooling rate ($\gamma =$)	0.8	0.9	0.99
$T_{min} = 0.5$	199.22	213.97	238.51
$T_{min} = 5$	186.22	194.96	216.74

Table 2: Performance with different values for Simulated Annealing parameters. The performance is reported for the FanDuel dataset, using the 'Average' projection scheme.

	Average	Big3	Adhoc	Fully	Average	Big3	Adhoc	Fully
Average	186.32	204.22	208.48	238.51	186.32	204.22	234.20	243.55
Last	175.95	210.83	218.58	248.49	175.96	210.84	243.00	245.50
Regression	192.03	208.78	218.39	241.45	192.03	208.78	244.72	244.30

Table 3: Performance (measured in total actual fantasy points) of different line-up generating methods (columns) with different projection methods (rows). The left 4 columns are tested on data from FanDuel, while the right columns are on DraftKings.

use $\gamma = 0.99$ and $T_{min} = 0.5$ for the Simulated Annealing solver when comparing against other methods. To randomly perturb between different states, we replace an randomly chosen player from our line-up with a player from the available pool, who can satisfy both the salary cap and position constraints.

7 Experiment Setup and Results

Our data contains 35 days of NBA fantasy data. As discussed in previous section, we use the first 20 days for training and the last 15 days for evaluations. For each of the last 15 days, we compute the actual fantasy points total for each strategy's line-up. We then average these values to obtain the final comparison metrics.

Table 3 shows the performance on both FanDuel and DraftKings dataset using different projection methods and line-up generation strategies. We observed the following. (1) Accurate projections do not mean better final solutions. Even though the 'Last' projecting method have the highest errors, as shown in Table 1, optimization with this projection yield the best mean actual fantasy points. (2) Factoring the position constraints into the formulation significantly improves the results in FanDuel dataset and yield a slight increase in DraftKings dataset.

As a point of reference, the winning total for a line-up in the beginners' bracket is around 250. While our best result is still below the expected winning cut-off, it is close. Furthermore, our system ignores many important factors, such as injury reports, opponent ranking, home or away game.

8 Conclusion

In this project, we aim to automatically generate a lineup for a daily fantasy basketball contests. We experiment with different fantasy point projection methods. Using the projected fantasy points, we compare the performance of different formulations. Our best is just a bit short of a winning cutoff in the beginner contests.

References

Kasten, T. (2009, November 17). *System and method for using draft position information to aid player selection in a fantasy league draft.* Google Patents. (US Patent 7,618,312)

Krieger, F. M., & Wirtschafter, A. (2008, April 28). *Predictive modeling system and method for fantasy sports.* Google Patents. (US Patent App. 12/111,054)

Morris, D., Krieg, D., Willmarth, D., & Kievatt, D. (2005, February 15). *System and method for predicting performance of fantasy athletes.* Google Patents. (US Patent App. 11/058,079)

Ware, S. B., & Webb, B. A. (2006, November 29). *Evaluation of fantasy players.* Google Patents. (US Patent App. 11/606,717)