

# **Fantasy Baseball Trade Suggester**

By: Devin Bhushan | Advisor: Prof. Luke Olson

9-May-2014

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Definitions</b>	<b>4</b>
<b>3</b>	<b>Introduction</b>	<b>5</b>
<b>4</b>	<b>Technologies Used</b>	<b>6</b>
	MongoDB . . . . .	6
	MongoHQ . . . . .	6
	Python . . . . .	6
	Github . . . . .	6
	Munkres - Python Library . . . . .	7
<b>5</b>	<b>Workflow</b>	<b>8</b>
	Collect Data . . . . .	8
	Import Settings . . . . .	8
	Compute Projected Points . . . . .	8
	Compute Optimal Lineups . . . . .	8
	Points Above Average (PAA) . . . . .	12
	Identify Strengths and Weaknesses . . . . .	15
	Suggest Trade Targets . . . . .	15
<b>6</b>	<b>Conclusion</b>	<b>18</b>

# 1 Abstract

*Forbes* reported that as of 2009, there were 11 million users for Fantasy Baseball. Since then, that number has only increased as mobile and web services have become more prominent. However, one pattern that most Fantasy Baseball users have noticed and reported is that the frequency of trades in their leagues is stagnant. Some leagues incentivize trades by allowing inclusion of draft picks in future years. However, by and large, trade frequency is low and this results in a less eventful fantasy experience. This dearth of trades can be traced back to the complicated nature of identifying a trade partner.

Before a trade is proposed, one has to explore the league to find a team that has a relative surplus at a position of one's need as well as a relative need at a position of one's surplus. Then there is the technical process of suggesting a trade that is statistically fair for both sides as well. There is a lack of a simple tool to assess a Fantasy Baseball league and identify both a trade partner and also a suggested framework (in terms of players involved) for said trade.

This paper proposes the foundations to such a tool; a tool that takes as input a Fantasy Baseball league and outputs trade suggestions for a given owner.

## 2 Definitions

**Sabermetrics** - A term derived from SABR (Society for American Baseball Research), referring to the empirical analysis of baseball and baseball statistics.

**NoSQL (“Not Only SQL”)** - A database that is modeled as a key-value store, where instead of being stored in tables with columns, documents are without relations in the form of pairs. This model for a database allows for easy access if data is indexed by numerous unrelated keys.

**Fangraphs.com** - A website that provides statistics for every player in MLB history. The website keeps track of every event of every game - including pitch location, type, speed, release point as well as batter details. Fangraphs also contains an editorial section with detailed statistical analysis of baseball.

**Moneyball** - *Moneyball: The Art of Winning an Unfair Game* is a book by Michael Lewis, it details the Oakland Athletics’ General Manager Billy Beane’s statistical (sabermetric) approach to building a winning baseball team despite a low budget.

### 3 Introduction

I have played Fantasy Baseball since I was just 14 years old. Over the years, I have developed some techniques for success in this stimulating virtual sport. Before the start of the 2006 baseball season, I read the book *Moneyball* and was first exposed to the Sabermetric world. The book was just the tip of the iceberg for me, I engulfed myself in the statistical haven that was Fangraphs.com and rabidly studied statistical trends and metrics.

Over the past few seasons, I have found myself having a hard time analyzing trades that would be beneficial for both my team and a potential partner. It is very time consuming to look through a whole league's worth of teams (somewhere between 8 and 12) and finding a valuable deal that would be worthwhile for both teams. Often this results in very few trades taking place in the league. While this is a consequence most leagues have accepted and are complacent with, I think it would add a whole new dynamic of competitiveness to the game if a piece of software existed that could not only algorithmically identify trade partners for your team, but also suggest which positions each team needs and might want to trade away.

Therefore, the idea for this project was born. The goal for it is to eventually be put into production such that it can hook into an active Fantasy Baseball league, a data source for statistical projections and provide a user with potential trade partners and specific trade suggestions that would be beneficial for both teams. While trades are generally subjective in nature, this paper serves as a means of analyzing trades through the lens of statistical prowess that the Sabermetric community has only recently been polishing.

## 4 Technologies Used

### MongoDB

MongoDB is a document-based NoSQL database. It interfaces very easily with Python and is one of the fastest databases currently in use. Storing and reading data from MongoDB is exceedingly easy, Python's primary data structure is a dictionary and dictionaries directly translate to the JSON format that MongoDB expects as input and provides as output. A key-value store is perfect for this application since it's primary storage is just league and player performance data, which is indexed by *league\_id* and *player\_id* - resulting in  $O(1)$  lookups.

Source: <http://www.mongodb.org/>

### MongoHQ

MongoHQ is a fully-managed platform for hosting MongoDB databases. It provides an easy interface to set up an instance of MongoDB for the application to use.

Source: <http://www.mongohq.com/>

### Python

Python is a powerful and fast programming language that allows for easy prototyping and interfaces well with all other technologies being used for this project.

Source: [www.python.org](http://www.python.org)

### Github

Github is a web-based hosting service for Git-style version control for this project.

Project Source: [https://github.com/devinbhushan/FFBaseball\\_Trade\\_Suggester\\_Thesis](https://github.com/devinbhushan/FFBaseball_Trade_Suggester_Thesis)

## Munkres - Python Library

The Munkres python module provides an implementation of the Munkres algorithm. It models an assignment problem as an  $O(N \times M)$  cost matrix, where each element represents the cost of assigning the  $i^{\text{th}}$  worker to the  $j^{\text{th}}$  job, and it solves for the least-cost solution. This algorithm proves essential in computing the optimal lineups for all teams.

Source: <https://pypi.python.org/pypi/munkres/>

## 5 Workflow

The current application uses a mock source of baseball projections as well as a mock Fantasy Baseball league from 2010. The roster and scoring settings are those imported from a standard 10-team, Head-to-Head points-based league.

### Collect Data

In order to allow future development of a plug-in for online fantasy leagues, this application is developed such that a league can be imported into the database. Thus, data from the database is standardized and the source of the data - online or offline - becomes irrelevant. Conveniently, NoSQL does not require players to be indexed by specific columns - collision of name, position, and team is a common problem when storing a large amount of athletes in a relational database. Therefore, a JSON object can easily be created for a player and his statistics to be stored in the database.

The sample statistical projections for all players are also loaded into the database in a similar manner. In the future, a module can be used that reads from a dynamic source of data for projections - allowing for more robust trade analysis.

### Import Settings

Every fantasy league has unique scoring settings as well as roster settings. Therefore, there is an additional plug-in system to allow to import these settings from an online league. Currently, a sample league's settings are being utilized.

### Compute Projected Points

Each player's projected point total is computed by using a league's scoring settings (that specify point distribution) in tandem with the player performance projections.

### Compute Optimal Lineups

Computing optimal lineups is the first step to determining strengths and weaknesses for a team, which itself is a stepping stone to finding an appropriate trade partner. Performing



this computation is actually a difficult process since a team may have multiple players who are eligible for multiple positions.

For example, **Figure 5.1** demonstrates a group of players that play a common set of positions. It is difficult to determine the configuration of positions and players that results in the optimal lineup, as per projections. **Figure 5.2** shows one such optimal lineup. However, analyzing the lineup in **Figure 5.1** and arriving at a solution like that in **Figure 5.2** requires one to find the optimal matching between players and positions such that the projected point total is maximized.

Player	Eligible Positions	Projected Points
Joe Mauer	C, 1B, U	540
Buster Posey	C, 1B, U	573
Brandon Belt	1B, OF, U	497
Jose Abreu	1B, U	467
Miguel Cabrera	1B, 3B, U	615

Figure 5.1: Potentially Conflicting Lineup

Player	Optimal Position	Projected Points
Joe Mauer	1B	540
Buster Posey	C	573
Brandon Belt	OF	497
Jose Abreu	U	467
Miguel Cabrera	3B	615

Figure 5.2: Optimal Lineup

This problem can be represented as a Bipartite Matching where each player on the team can be represented as a node on one side of the graph and each position in the lineup can be represented as a node on the other side. Positional eligibility is reflected on this graph as an edge connecting the player and the position. The edge weights are set to the projected point total for the corresponding player. **Figure 5.3** illustrates this process for our current example.

Now, ideally, running a simple Maximum Bipartite Matching algorithm on this graph can produce **Figure 5.4** such that every position has a player mapped to it (when possible)

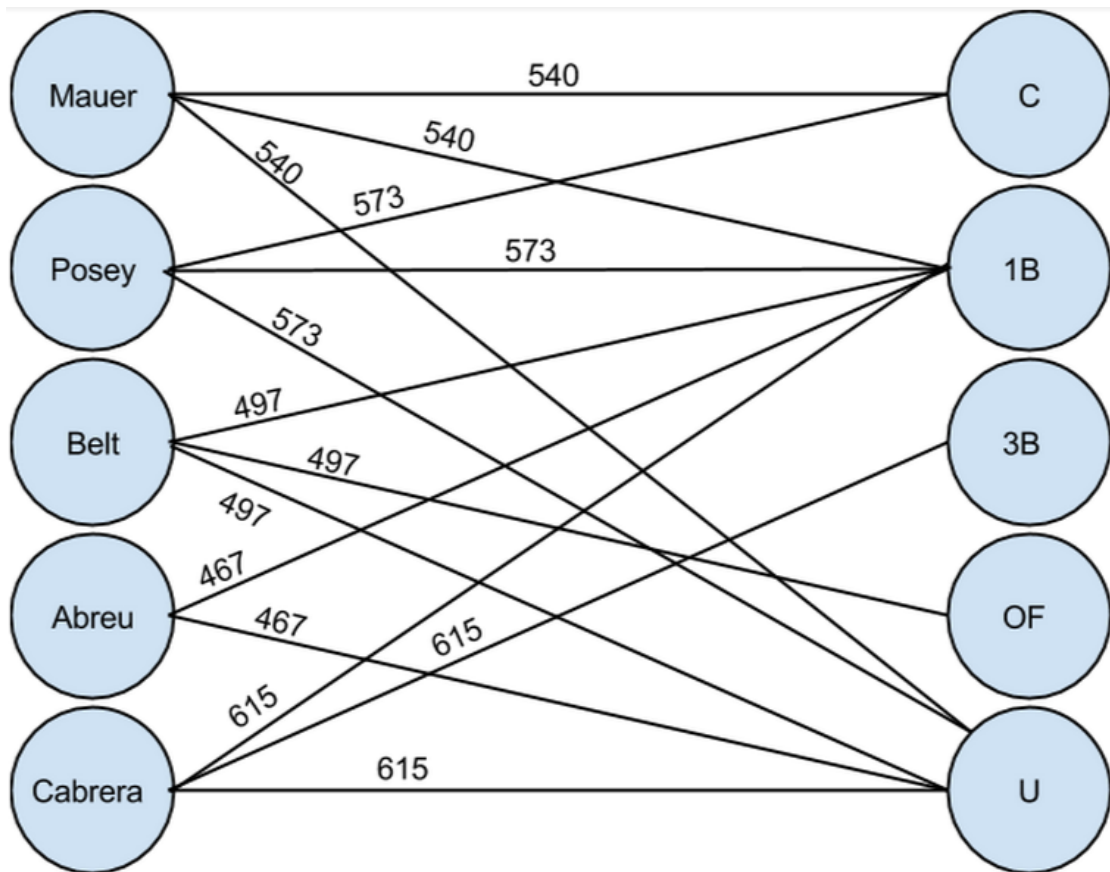


Figure 5.3: Lineup possibilities in bipartite graph

and the total number of points contributed to all positions is the maximum amount possible from all permutations.

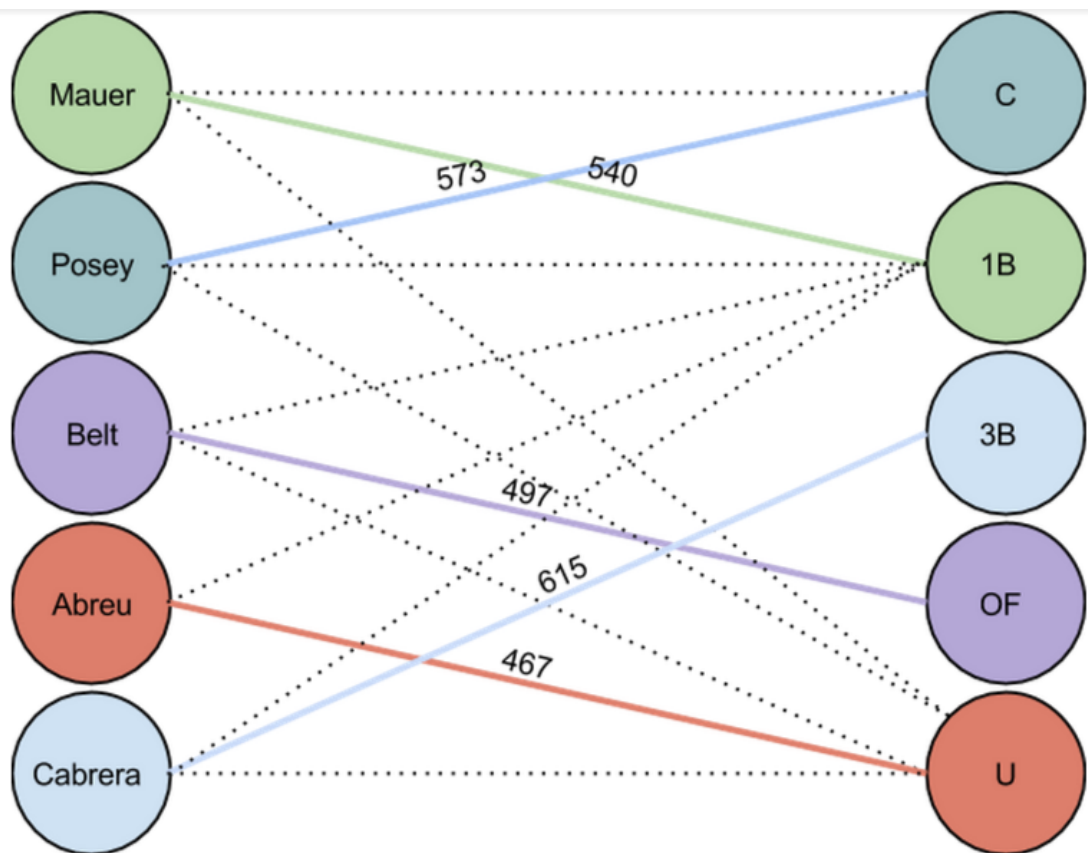


Figure 5.4: Optimal lineup using maximum bipartite matching

However, the easiest implementation of a Max-Flow, Bipartite Matching, or similar algorithm happens to be Munkres Assignment Problem (also known as The Hungarian Algorithm<sup>1</sup>). This approach to the problem results in an  $O(N \times M)$  running time, in terms of  $N$  players and  $M$  positions - therefore, it does not scale well for large input sizes<sup>2</sup>. However, within the scope of a fantasy baseball team, the input size is a constant (about 20 players and 10 teams) so the subpar performance is never exposed. This solution will scale for any league since the largest, most eccentric, leagues still have a small, finite upper bound on the number of teams and number of players per team.

This algorithm models an assignment problem as an  $N \times M$  cost matrix, where each element represents the cost of assigning the  $i^{\text{th}}$  worker to the  $j^{\text{th}}$  job, and it solves for the least-cost solution. In order to represent a fantasy team along with possible positions as an assignment problem, each column in this  $N \times M$  matrix represents a player on the team

<sup>1</sup>[http://en.wikipedia.org/wiki/Hungarian\\_algorithm](http://en.wikipedia.org/wiki/Hungarian_algorithm)

<sup>2</sup><https://pypi.python.org/pypi/munkres/>

while each row represents a position the team needs to fill. Each element,  $(i, j)$ , in this matrix represents the number of projected points player  $i$  could potentially contribute to position  $j$ . Ineligibility for a position is represented by a 0.

However, Munkres' algorithm solves for a least-cost solution. Each player's projected points are subtracted from the maximum representable integer. Therefore the least-cost solution actually selects the maximum-point lineup while satisfying all positional eligibility conflicts.

	<b>C</b>	<b>1B</b>	<b>3B</b>	<b>OF</b>	<b>U</b>
<b>Joe Mauer</b>	max-540	max-540	max	max	max-540
<b>Buster Posey</b>	max-573	max-573	max	max	max-573
<b>Brandon Belt</b>	max	max-497	max	max-497	max-497
<b>Jose Abreu</b>	max	max-467	max	max	max-467
<b>Miguel Cabrera</b>	max	max-615	max-615	max	max-615

Figure 5.5: Sample inverted-cost assignment matrix to find optimal lineup

The previous sample lineup of players is illustrated in **Figure 5.5** in a matrix appropriate to be used for Munkres' assignment problem. Solving this case by hand would require all permutations of this matrix such that no row and column are used more than once.<sup>3</sup> Then the matrix corresponding to the permutation which produces the least amount of points (since it is an inverted matrix) would be the solution. In this case, there are multiple solutions that are considered optimal so any of them are acceptable. Running the assignment problem on this problem gives us the indexes of elements which belong in a lineup that belongs to the set of optimal lineups. **Figure 5.6** outlines this process for the sample case.

## Points Above Average (PAA)

Once optimal lineups have been calculated, the projected performance is known for every team in the league. This makes the calculation of the expected contribution at every position possible by taking an average of projected starters for every team and every position. Using this positional average, the number of Points Above Average (PAA) can be computed for every player in the league for every position the player contributes

---

<sup>3</sup>The Munkres module used for this project automatically takes rectangular matrices, like the ones produced by fantasy baseball teams (since there are more players on a team than positions), and converts them to square matrices by adding Null value rows or columns. Thus, the one row and one column rule still applies.

<b>Optimal Lineup Indexes</b>	<b>Points</b>
(1,0)→Joe Mauer @ 1B	540
(0,1)→Buster Posey @ C	573
(3,2)→Brandon Belt @ OF	497
(3,4)→Jose Abreu @ U	467
(2, 4)→Miguel Cabrera @ 3B	615
<b>Total Points</b>	<b>2692</b>

Figure 5.6: Solution indexes and corresponding projected points

points to. This allows multi-positional eligibility to be assessed in context. The positional averages for a sample league from 2010 are listed in **Figure 5.7**. For example, a player like Miguel Cabrera, who we previously saw is projected at 615 points, who is eligible at 1B, 3B, and U is most valuable when used at 3B since that position has the lowest average of them all. The PAA metric is reflective of that fact.

<b>Position</b>	<b>Avg. Points</b>
C	304
1B	497
2B	454
3B	473
SS	439
OF	499
U	566
SP	425
RP	179

Figure 5.7: Expected Starting Player Positional Avg Contributions

## Identify Strengths and Weaknesses

Each team's strengths and weaknesses need to be identified before a trade suggestion can be made. The PAA statistic is very useful in determining strengths and weaknesses of a particular team. A team's relative strengths and weaknesses are easily determined by examining each player's PAA. A positive PAA value reflects an above league average and this player is considered a strength relative to the rest of the league. Vice versa for negative PAA values. It is possible that a team could have 2 players who only play the same position and while both have positive PAA, only one of them can start in the lineup. This is deemed a surplus and is easily identifiable algorithmically with the help of PAA. Additionally, by adding "Bench" as a few positions in the lineup, the optimal bench can be determined as well. This allows for identification of a weak (or strong) bench, potentially allowing for future features that focus on upgrading a bench by looking at the free agent pool.

## Suggest Trade Targets

Using the aforementioned strengths and weaknesses of every team, it seems fairly simple to match up one team's strengths with another's weaknesses and call them trade partners. However, trades are complicated in that not only do both teams prefer to deal from strengths to quench weaknesses, but they also need to be assured that the outgoing player value is close, if not equal, to the incoming player value. This situation is witnessed in the trade featured in **Figure 5.8**. In this case, Team A is actually "winning" the trade if only the raw net change in points is considered. But on further inspection using the PAA statistic (and accounting for positional differences), it becomes clear that while David Ortiz is projected significantly higher than Brian McCann in raw points, he is actually less valuable after factoring in that he is only eligible for U (the position with the highest positional average, see **Figure 5.7**) and that Brian McCann plays the position with the lowest average and vastly outproduces it. Thus, in determining "fair" trades, using overall net PAA change is actually a far better metric than just raw points projected.

In order to further illustrate scenarios where the trade suggester proves useful, **Figure 5.9** and **Figure 5.10** show two trades that are unique in different aspects.

The trade discussed in **Figure 5.9** is unique because at first glance it seems as if Team A is giving up far too much - in raw points, Team A gives up 274 more points than it gains. However, in terms of PAA, the trade is remarkably fair (only a 2 point difference) and if it addresses each team's needs then it's worth it for both teams.

The trade from **Figure 5.10** exemplifies why this tool is so effective, trades involving numerous players (especially those of different positions) become easy to evaluate and find. In this case, intuitively Team A should accept that trade because it is receiving over 2200 raw points while only giving up about 750. But once again, the situation changes

Team A receives

Player	Position	Projected Points	PAA
Brian McCann	C	500	(+)196

Team A gives up

Player	Position	Projected Points	PAA
David Ortiz	U	549	(-)17

Figure 5.8: Positional PAA difference in trades

Team A receives

Player	Position	Projected Points	PAA
Brian McCann	C	500	(+)196
Chad Billingsley	SP	390	(-)35

Team A gives up

Player	Position	Projected Points	PAA
David Ortiz	U	549	(-)17
Troy Tulowitzki	SS	615	(+)176

Figure 5.9: A deceptively fair trade with PAA



Team A receives

Player	Position	Projected Points	PAA
Aaron Hill	2B	460	(+)6
Asdrubal Cabrera	SS	449	(+)10
Billy Butler	U	530	(-)36
Ervin Santana	SP	467	(+)42
Wilson Ramos	C	314	(+)10

Team A gives up

Player	Position	Projected Points	PAA
Max Scherzer	SP	613	(+)188
Javier Lopez	RP	135	(-)44

Figure 5.10: A deceptively unfair trade with PAA

drastically once PAA and positional eligibility is considered - Team A is only receiving about 32 PAA while it is giving up about 144.

## 6 Conclusion

Trades, it turns out, are a difficult thing to measure objectively. Outside of projections and statistics, there can be attachment (or lack thereof) to players that keeps owners from pulling the trigger on specific trades. Sometimes, owners have hunches that they like to play out. After all, every year there are numerous players that blossom out of, seemingly, nowhere while some inexplicably bust.

This application attempts to bring a statistical perspective to fantasy baseball trades. The frequency of trades has always been low in this virtual sport. The challenge for owners is to find a suitable trade partner, come up with an acceptable trade for both parties, and convince the other owner to accept the deal. When talks don't work out, the entire process needs to be repeated from the beginning. Often the typical fantasy owner has a job or occupation that prevents them from spending extra time investigating potential fantasy trades.

For example, the trades in **Figure 5.9** and **Figure 5.10** are remarkably deceptive unless they are heavily analyzed. This tool only begins to scrape the surface of possibilities for analyzing fantasy baseball through a sabermetric lens. There are a number of things standing between this prototype and a product. Notably, currently it only works with an offline feed of data. In order to be released to the public, plug-ins are needed to feed data in from Yahoo! Fantasy leagues to dynamically suggest trades over the course of a season.

With more time, I would have put more thought into projecting injuries. Specifically, being able to adjust the positional averages in a more intelligent manner than just taking an average of all starting players. This requires research into injury frequency and projected replacements from the waiver wire.

Additionally, this tool can be expanded to manage the waiver wire by monitoring free agents over the course of the season and helping to optimize a team's lineup.

However, the goal of this project was to build a prototype that proved it was possible to intelligently analyze a fantasy baseball league and suggest trades that were mutually beneficial - an idea that is thought to be a myth in the land of fantasy baseball.