

# Building an Optimal Roster for Rotisserie Baseball

## Introduction

Three years ago my roommate invited me to participate in his 10-team 5x5 rotisserie baseball league. I played baseball growing up, but never had much interest in the MLB. I accepted the offer to join – after all, \$10 is a small price to pay for a reason to watch baseball and an opportunity to glorify the part of baseball that matters the most - the stats. The thing that's intrigued me the most has always been the draft – we use an auction draft, which adds an extra dimension to the strategy. With no set draft positions, there's no comfort of the ability to take the best player available. Every pick is fought for, and to draft well requires a coherent strategy.

That first year, I had a relatively naive strategy: a list of a few players at each position that I wanted and a long list of players I didn't. I didn't put much thought into my budget, so my lists were exhausted midway through the draft. The second year, I didn't improve my strategy much...I think. Try as I might, I cannot remember anything about this draft. Every other draft I remember vividly, but not this one. I was about to graduate college, there was a lot going on – no room in my head for rotisserie drafts I guess. Last year, with the league expanding to 12 teams, I improved my strategy a bit. I started by coming up with goal values for each category based on previous winning values. I exported Steamer/Razzball projections into an excel sheet, set up to calculate my team's score for each category and tell me what percentage of my goal I'd reached so far. I knew I needed to step up my strategy for this year.

At its most basic, we can think of a draft as an optimization problem, albeit with a dizzying number of unknowns. But, if we assume that our Steamer/Razzball projections and valuations are basically correct – and make a few other simplifications – we can formulate our draft as a basic maximization problem that we can solve to determine the best overall roster available through the draft.

## Problem Formulation

We start by formulating a linear equation for our roster's overall score:

$$S = \sum_{i \in Hitters} H_i^{adj} + R_i^{adj} + HR_i^{adj} + RBI_i^{adj} + AVG_i^{adj} + \sum_{j \in Pitchers} W_j^{adj} + K_j^{adj} + SV_j^{adj} + \frac{1}{ERA_j}^{adj} + \frac{1}{WHIP_j}^{adj} \quad (1)$$

There are a few things to note here. Each of these is an adjusted value. Since each category has a different magnitude but each is worth the same number of points, we'll need to adjust them. Because we want each category to have the same scale but maintain each player's relative value in each, we'll need to

center their means but preserve their variances. We'll use the sklearn Standard-Scaler method to do so (specifying to maintain the standard deviation). I've used average, ERA, and WHIP directly in this equation, despite the fact that they're actually calculated overall for the entire team from the base statistics. I experimented with separating them out, but (somewhat suprisingly) found no added value in doing so. For clarity, I've split the pitchers and hitters into two different summations, but we will end up combinining them into one. This will be equivalent to equation (1) since hitters do not generate pitching stats and pitchers do not generate hitting stats.

$$S = \sum_{p \in Players} H_p^{adj} + R_p^{adj} + HR_p^{adj} + RBI_p^{adj} + AVG_p^{adj} + \quad (2)$$

$$W_p^{adj} + K_p^{adj} + SV_p^{adj} + \frac{1}{ERA_p}^{adj} + \frac{1}{WHIP_p}^{adj}$$

So, now that we have a single equation to calculate our score, we need to find the group of players that maximize it within the constraints of the league.

$$Team = \operatorname{argmax}_{s \in S} \sum_{p \in s} H_p^{adj} + R_p^{adj} + HR_p^{adj} + RBI_p^{adj} + AVG_p^{adj} + \quad (3)$$

$$W_p^{adj} + K_p^{adj} + SV_p^{adj} + \frac{1}{ERA_p}^{adj} + \frac{1}{WHIP_p}^{adj}$$

$$s.t. \sum_{p \in s} c(p) \leq 260 \text{ where } c(p) \text{ is the cost of player } p$$

$$\{C, 1B, 2B, 3B, SS, 1B/3B, 2B/SS, 1B/2B/SS/3B, 6OF, 8SP, 3RP\} \in s$$

$$\sum_{p \in s} N(p) = 25 \text{ where } N(p) = \begin{cases} 1 & \exists! n \in s (n \text{ is name of player } p) \\ 0 & \text{else} \end{cases}$$

where S is the set of sets of all possible combinations of 25 players

## Results & Conclusion

Using Gurobi, we can solve equation (3) to find our optimal set of players.

Table 1: Hitters

Name	Position	R	HR	RBI	SB	AVG
Jorge Polanco	SS	94.6	19.3	84.7	7.6	0.281
Jose Abreu	1B	84.5	31.1	100.4	2.6	0.276
Nicholas Castellanos	OF	87.7	26.9	88.4	3.0	0.274
Bryan Reynolds	OF	83.1	17.9	66.0	6.3	0.288
J.T. Realmuto	C	77.8	24.0	76.4	5.8	0.270
Matt Chapman	3B	94.2	34.7	98.2	2.7	0.258
Paul DeJong	SS	82.0	28.4	84.9	6.8	0.253
Eric Hosmer	1B	79.7	23.4	80.2	3.4	0.263
David Peralta	OF	80.6	20.6	74.4	2.5	0.279
Franmil Reyes	OF	82.1	37.0	101.5	1.3	0.260
Jason Heyward	OF	72.0	15.8	72.1	6.5	0.261
Shin-Soo Choo	OF	83.3	21.0	61.6	9.0	0.249
Jonathan Schoop	2B	67.4	26.5	77.0	2.3	0.262
Brandon Lowe	2B	70.9	23.7	71.6	6.7	0.246
Total		1139.9	350.3	1137.4	66.5	0.267

Table 2: Pitchers

Name	Position	W	SV	K	ERA	WHIP
Gerritt Cole	SP	14.9	0.0	273.3	3.250	1.040
Trevor Bauer	SP	11.6	0.0	228.4	3.920	1.240
Luis Castillo	SP	11.3	0.0	203.4	3.820	1.250
Lucas Giolito	SP	10.8	0.0	198.4	4.240	1.270
Matthew Boyd	SP	10.0	0.0	194.2	4.320	1.220
Robbie Ray	SP	9.9	0.0	197.5	4.120	1.320
Caleb Smith	SP	8.2	0.0	172.4	4.580	1.320
Mike Foltyniewicz	SP	9.7	0.0	157.5	4.70	1.350
Joe Jimenez	RP	2.8	20.0	69.1	4.150	1.260
Hansel Robles	RP	3.3	20.0	70.9	4.180	1.270
Jose Leclerc	RP	3.5	20.0	91.9	3.390	1.210
Total		96.0	60.0	1857.0	4.072	1.246

Then, we can use the scores and rankings from last season (with my team removed) to get a score for our team. The category and overall rankings are as follows:

Table 3: Category Rankings

Rank	R	HR	RBI	SB	AVG	K	W	SV	ERA	WHIP
1	1241	433	1198	141	0.2768	1788	112	115	3.493	1.102
2	1174	382	1147	127	0.274	1725	109	105	3.665	1.131
3	1159	355	1088	123	0.2735	1643	99	79	3.76	1.21
4	1136	353	1077	121	0.271	1598	98	73	3.898	1.216
5	1110	352	1075	113	0.2705	1531	97	59	3.907	1.244
6	1067	332	1030	110	0.2645	1480	93	59	4.107	1.247
7	1006	321	1016	106	0.2642	1391	85	55	4.112	1.262
8	997	302	1000	97	0.2641	1387	84	54	4.144	1.267
9	974	291	955	94	0.262	1336	80	42	4.217	1.284
10	966	284	905	73	0.2601	1330	74	39	4.444	1.291
11	898	260	897	72	0.2592	1132	64	3	4.616	1.339

Table 4: Overall Rankings

Rank	Points
1	95.5
2	83
3	74
4	74
5	73
6	70
7	69.5
8	57
9	51
10	45.5
11	24

We can obtain scores for each of our values by comparing them to our table of results for last season. As a refresher, rotisserie baseball leagues are scored by combining each of the category for the whole team, comparing all teams for each category, and granting 12 points to the first place team, 11 points to the second, etc.

Table 5: Team Score

Value	Points
R	9
HR	5
RBI	9
SB	5
AVG	7
K	12
W	7
SV	8
ERA	7
WHIP	7
Total	76

All things considered, this is not a bad draft - if we refer to table 4, this puts us in 3rd place in the league. However, if we were to accept that the draft is a simple linear optimization problem, we would expect that our team would be in first place. So, why isn't it? We could conclude that simply drafting the perfect team is not good enough to win a rotisserie baseball league - that drafting perfectly could get you a very good team, but playing the waiver wire is required to come in first place. I think this is probably true, but I think there's something else that it's very important we do not overlook: **a rotisserie baseball auction draft cannot actually be represented by a linear optimization problem.** Once your team gets to first place in any particular category adding to that category further gives you no additional value. Nothing is gained by running up the score in any category, but our model doesn't represent this. To truly model the draft, we'd need a **stepwise** optimization. We'd need to apply our scoring function to our model (and, of course, come up with a new way to score our model - perhaps by applying it in a series of mock drafts). However, this increases the complexity of solving our optimization an incredible degree. Despite this limitation, our model does very well at approximating an optimal rotisserie baseball team for our particular league. Another thing that is important to note is that this model depends completely on the accuracy of the Steamer/Razzball predictions. Despite using these predictions for a number of years, I have not analyzed their accuracy.

## Appendix