Lineup Optimization

A Markov Chain Approach to MLB Batting Orders

Neil Patel, Shayan Haque, Andy Nguyen, Chenzhou Li

Mentored By: Anton Daburah

JOHNS HOPKINS
WHITING SCHOOL *of* ENGINEERING

Introduction

We present a full-stack lineup optimization system that models an MLB batting order as a Markov chain over twenty-four base-out states. Each batter's contribution is quantified through Baserunner-Dependent Net Run Production (BRP) - a four-batter window metric that captures the context-dependent value missed by traditional statistics such as OPS or wRC+, An exhaustive search over all 9! = 362,880 permutations (accelerated with Numba JIT, under two seconds) identifies the optimal batting order, which is then compared against actual manager decisions across the full MLB schedule.

Our Method

1 MARKOV STATE SPACE

Twenty-Four Base-Out States

Eight base configurations across three out counts. When the third out is recorded, the inning ends and the chain enters an absorbing state.

0 out	ø empty	1 1st	2 2nd	12 1&2	3 3rd	13 1&3	23 2&3	L loaded
1 out	ø empty	1 1st	2 2nd	12 1&2	3 3rd	13 1&3	23 2&3	L loaded
2 out	ø empty	1 1st	2 2nd	12 1&2	3 3rd	13 1&3	23 2&3	L loaded
3 out	■ ABSORBING STATE ■ inning ends							

Each batter's transition probabilities are derived from counting statistics — **out rate, 1B, 2B, 3B, HR, and BB/HBP** — and are further split by a **speed factor** (normalized log-scaled stolen-base rate) into fast- and slow-runner variants.

2 THE FOUR-BATTER WINDOW

Context-Dependent Chaining

A → B → C → D

A: leadoff; B: conditioned on A; C: conditioned on A, B; D: conditioned on A, B, C

Starting from **bases empty, zero outs**, each batter expands into every possible base-out state, weighted by their transition probabilities. Runner advancement depends on the speed factors of preceding batters.

$$T[i, j, k, l] = BRP(batter_i, batter_j, batter_k, batter_l)$$

→ A 9 × 9 × 9 × 9 BDNRP tensor

The tensor encodes the expected net run production for every ordered quadruple of batters, forming the search space for optimization.

3 BRP FORMULA

Run Expectancy Valuation

Each terminal base-out state is valued using the **RE24 matrix**, which gives the expected runs from any state through the end of the inning.

$$BRP_{state} = (runs + RE24[new] - RE24[start]) \times P(state)$$

$$total\_BRP = \Sigma\, BRP_{state} \quad \textit{over all terminal states}$$

RE24 MATRIX (EXPECTED RUNS)

	ø	1	2	12	3	13	23	L
0 out	0.48	0.86	1.10	1.44	1.35	1.78	1.96	2.29
1 out	0.25	0.51	0.66	0.88	0.95	1.13	1.38	1.54
2 out	0.10	0.22	0.32	0.43	0.35	0.48	0.58	0.75

BRP decomposes into **GOB** (get-on-base) and **PUDI** (push-up / drive-in) components for visualization and diagnostic analysis.

Results

Actual vs. Optimal Batting Order

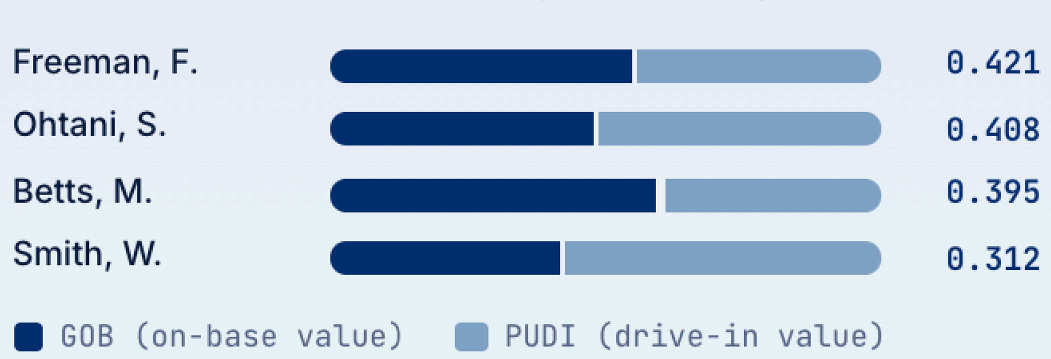
ACTUAL MANAGER ORDER

1	Betts, M.	0.412
2	Freeman, F.	0.389
3	Ohtani, S.	0.361
4	Muncy, M.	0.298
5	Smith, W.	0.276

OPTIMAL ORDER (BRP)

1	Freeman, F.	0.421
2	Ohtani, S.	0.408
3	Betts, M.	0.395
4	Smith, W.	0.312
5	Muncy, M.	0.289

PER-BATTER BRP BREAKDOWN (GOB + PUDI)



+0.09
Δ RUNS / CYCLE

Typical MLB managers leave **0.05–0.15 expected runs per cycle** on the table relative to the BRP-optimal order, largely because of conventional heuristics (best hitter batting third, pitcher batting ninth).

Optimization Engine

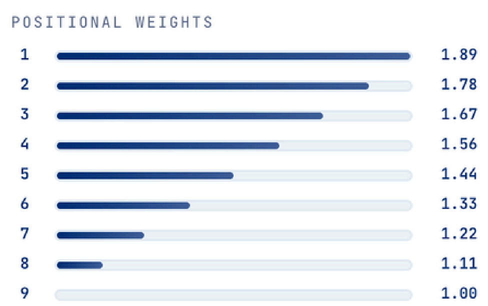
Position-Weighted BRP Across the Order

The lineup score uses a **sliding window of four** that wraps around the order (position 9 feeds into positions 1–3), capturing the cyclic nature of a batting order.

```
score = Σ T[order[(p-3) mod 9],
            order[(p-2) mod 9],
            order[(p-1) mod 9],
            order[p]] × w[p]

w[p] = 1 + (8 - p) / 9   for p = 0 … 7
w[8] = 1.0                9th spot
```

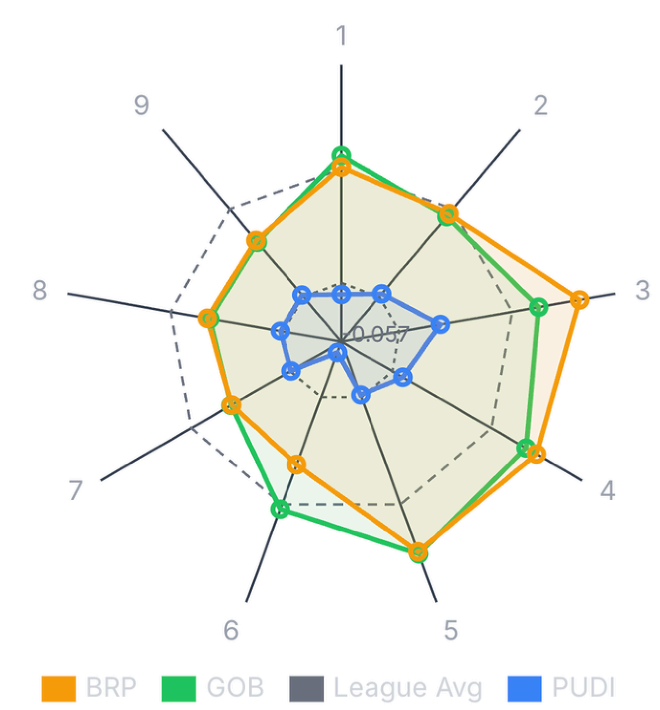
POSITIONAL WEIGHTS



Weights reflect the expected number of additional plate appearances a position receives over a full game.

Supported constraints: positional locks (pin a player to a slot), handedness limits (maximum consecutive LHB or RHB, with wraparound), and speed-based platoon splits.

362,880	< 2s	6,561
9! PERMUTATIONS	NUMBA JIT SEARCH	BDNRP TENSOR ENTRIES



Try Our Website!

Key Features

A Daily Lineups Comparison
Browse any MLB game day. View the manager's actual order alongside the optimal order, with radar charts (GOB / PUDI / BRP) and an expected-runs delta.

B Lineup Builder
Search any MLB player, configure platoon splits, drag and drop batting positions, apply handedness constraints, and export or share optimized lineups.

C Radar Charts & BRP Breakdown
A Recharts radar visualization of GOB, PUDI, and total BRP per batter, plus color-coded stat tables and per-player contribution bars.

D Game Archive & Lineup Hub
Calendar-based historical browsing backed by a PostgreSQL cache, with social lineup sharing, a request system, user comparison threads, and comments.

Build the lineup your team actually needs.