AN OPTIMIZATION OF BASEBALL FIELDER POSITIONING USING SEAM

BY

COLIN ALBERTS

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Applied Mathematics
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Adviser:

Professor Daniel Eck

Abstract

In the last decade, professional baseball has witnessed significant statistical and analytical advancements that have profoundly impacted on-field strategies. This study delves into diverse strategies for positioning defensive players by employing deterministic and stochastic gradient methods. These methods optimize fielder alignments based on a synthetic distribution estimating batted ball distributions across all Major League Baseball (MLB) batter-pitcher matchups in any MLB ballpark. The primary aim is to devise a fielder placement model that minimizes expected batting average for balls in play by maximizing the density coverage of each fielder. This research explores various optimization techniques to identify the most effective approach in terms of both optimality and implementation practicality. This methodology could be generalized to determine optimal fielder alignments for similar sports or applications such as softball, cricket, or region design. The code for this work can be found in this GitHub repository.

Table of Contents

1 Introduction

Baseball is a sport renowned for its rich history, intricate game planning, and the fusion of athleticism and strategy. The game is rooted deep in American tradition and cherished by fans worldwide, providing a captivating medium for players, casual fans, and statisticians alike. Although baseball's roots begin in America's northeast (Block [2]), the game has drawn worldwide popularity in terms of its participants and spectators. Countries outside of the United States such as Japan, Venezuela, the Dominican Republic, Cuba, and Puerto Rico have adopted the game and integrated into Major League Baseball (MLB) in doing so. Through this popularity, the game has evolved to address emerging philosophies, new technologies, and advancements in training methods that incur new aspects of the game.

At its core, baseball is a team sport played by two teams, each comprised of nine players playing at any given time. Although baseball is a team game, an individual aspect occurs when the offensive team's batter attempts to strike the ball thrown by the defensive team's pitcher into fair (playable) territory. Through each of these individual matchups, the defensive team seeks to generate outs, and hence prevent runs, through various actions such as catching the ball when it is hit in the air, tagging runners with the ball, forcing runners out at a base, or striking out the batter. It is the goal of the defense, commonly referred to as the fielders, to create these outs to prevent the offensive players from safely advancing the bases. The objective of the game is to score more runs than the opposing team with a prescribed number of innings, i.e. chances to score runs. There are many other rules and nuances to the game, but this outlines the core objective and provides a base understanding moving forward.

This writing focuses on positioning the nine defensive (fielding) players in a manner to seek outs. The pitcher is fixed on the pitcher's mound and the catcher is fixed behind home plate to catch the pitched ball. The other seven fielders are allowed to move about the in-play territory, with tradition placing four of the fielders near the bases and three fielders in the outfield. Fielder location is vital for defensive strategy as proximity to where the ball is hit dramatically affects the probability of a fielder making an out.

Examining the events of the game more closely, we can observe unique properties of a pitch or batted ball event from every player, giving greater insight into the player's strategy and approach to the game. For pitchers, the properties of each pitch include its velocity, release point, spin rate, and movement, among others. For batters, this may include how hard they hit the ball (commonly referred to as exit velocity), the angles at which the ball travels off of the bat (commonly referred to as launch angle and spray angle), and the position of the ball when it crosses home plate from the catcher's perspective. The MLB uses high-speed tracking cameras and software to collect every instance of play and publishes the data (and its aggregations) publicly (Major League Baseball [1]).

Access to such data has ushered in a new frontier of study regarding the distribution of batted balls and how teams can take advantage of this information to optimize their fielder positioning to generate more outs. This has led to MLB teams taking advantage of shifted defensive alignments to capitalize on a batter's batted-ball tendencies. For example, an extra infielder may be placed on the side of second base in which the batter is more likely to hit the ball. Teams conventionally used a "straight-up" alignment, one where there are two infielders on each side of second base with three players spread across the outfield. This is becoming increasingly uncommon among MLB teams with the rise of data-driven fielding insights and statistical optimization techniques built to streamline the fielder placement process. During our investigation the MLB implemented a rule change to limit the scope of defensive shifting; this can be incorporated by adding fielder constraints.

Among these techniques, the Synthetic Estimated Average Matchup (SEAM) methodology (Wapner, Dalpiaz, and Eck [11]) combines the matchups between the empirical pitcher and batter, the empirical batter with a synthetic pitcher, and a synthetic batter with the empirical pitcher using convex combinations with weights describing the similarity of the synthetic players to the empirical players. Thus, this offers a mathematically justified spray chart for any given batter-pitcher matchup among MLB players since 2017. This attempts to solve the problem of sparse batter-pitcher matchup data by using statistically similar players as a proxy to calculate the expected batted ball distribution. Although the contents of this article will focus on the application of fielder placement in baseball, the constrained optimization techniques can be generalized to problems outside of the game. The SEAM distributions were selected for optimization as they offer a mathematically justified medium with a demonstrated ability to significantly reduce errors in comparison to other distributions.

Furthermore, we develop a fielder placement optimization methodology using SEAM distributions that arranges the seven non-pitcher and non-catcher fielders to optimally cover the field based on various maximization and minimization techniques. While previous studies have aimed to optimally place the three outfielders only (Murray, Ortiz, and Cho [7]) or base the alignment on traits of particular fielders (Grove [5]), I aim to use multi-objective spatial optimization models for all seven fielders to produce even more efficient defensive alignments. The goal of this work is to develop a fielder placement method that generates outs at a higher frequency than traditional and other data-driven techniques. I claim that this work is relevant as a 1% coverage increase implies about 40 additional outs each year per team; approximately one and a half full baseball games (Wapner, Dalpiaz, and Eck [11]). Thus, it is reasonable to maintain that software meant to optimize said coverage is valuable as MLB teams spend millions of dollars signing free agents, developing talent in their minor leagues, and paying lucrative contracts to existing players to achieve the same outcome: more outs and more wins.

The following section will dive into the details of SEAM, previously studied methods regarding fielder placement optimization in baseball, methods I implemented, and the technical aspects of tangential work done along the way.

2 Background

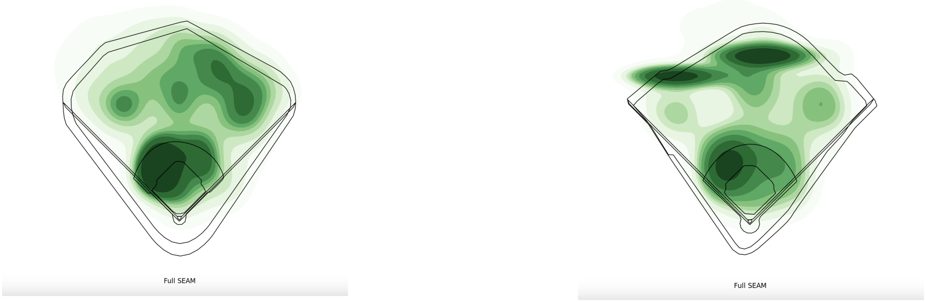
2.1 Previous Methods

2.1.1 SEAM

In the context of this problem, SEAM serves as the medium for the optimization, but the optimization is not necessarily specific to the SEAM application alone. This is simply one example of a constrained optimization technique over a multimodal distribution with 7 degrees of freedom in which the distribution is justified to be advantageous in context. Thus, it is possible to generalize these optimization techniques over smooth spatial distributions, but this article will observe these techniques through a baseball-specific lens in which constraints are added based on the context of fielder positioning. I chose SEAM as the medium because SEAM quickly provides batted-ball distribution estimates for individual matchups which 1) allows for many test cases for our techniques; 2) has demonstrated increased predictive accuracy over simpler techniques that do not use data from both the batter and pitcher in their estimation of a batted ball distribution (Wapner, Dalpiaz, and Eck [11]).

SEAM uses a multivariate kernel density estimator that combines the empirical smoothed distribution of batted balls for a batter-pitcher matchup with similar synthetic distributions that utilize similarity scores for the empirical batter and pitcher with similar batters and pitchers. This is done by choosing pitchers who have faced the empirical batter with similar pitch characteristics to the empirical pitcher to gather more data. A similar process is true concerning the batter; the main difference is that performance against pitcher characteristics acts as the comparison metric. SEAM then takes a convex combination of the empirical, synthetic batter, and synthetic pitcher distributions with intelligently chosen weights (see (Wapner, Dalpiaz, and Eck [11]) Appendix) to create an aggregate distribution that significantly lowers the mean-squared error (MSE) in comparison to a standard bivariate nonparametric Gaussian kernel density estimator. I will now define the aggregate distribution.

Allow n_e to represent the number of balls in play for the empirical batter-pitcher matchup. n_p represents an aggregate of similarity scores and the sample size of the number of balls in play for similar pitchers versus the empirical batter; n_b represents a similar metric but with similar batters versus the empirical pitcher. Let $\hat{f}_{\text{e}}(\mathbf{y}), \hat{f}_{\text{sp}}(\mathbf{y}), \hat{f}_{\text{sb}}(\mathbf{y})$ represent the smoothed kernel density estimates for the empirical matchup, synthetic pitcher matchup, and synthetic batter matchup, respectively. The convex combination of the distributions is defined as follows:



(a) Justin Verlander vs. Mike Trout

(b) Josh Hader vs. Nico Hoerner

Figure 1: Full SEAM Example Outputs

$$\hat{f}_{\smile}(\mathbf{y}) = \lambda_e \hat{f}_{\mathrm{e}}(\mathbf{y}) + \lambda_p \hat{f}_{\mathrm{sp}}(\mathbf{y}) + \lambda_b \hat{f}_{\mathrm{sb}}(\mathbf{y})$$

The constants $\smile_j$ are calculated as follows:

$$\lambda_e = \frac{\sqrt{n_e}}{\sqrt{n_e} + \sqrt{n_p} + \sqrt{n_b}}, \lambda_p = \frac{\sqrt{n_p}}{\sqrt{n_e} + \sqrt{n_p} + \sqrt{n_b}}, \lambda_b = \frac{\sqrt{n_b}}{\sqrt{n_e} + \sqrt{n_p} + \sqrt{n_b}}$$

SEAM also provides an abundance of test cases that range from relatively standard (Figure 1a) to greatly irregular (Figure 1b) by having access to every batted ball in the MLB since 2017. This allows for fine-tuning of model parameters as it contains almost any feasible batted-ball distribution. This was important in the context of the testing phase as it allowed for model generalization and gave a deeper understanding of the method's limitations.

2.1.2 Enhancing strategic defensive positioning and performance in the outfield

To address the issues associated with fielder positioning in baseball, various methods of spatial optimization have been deployed in an attempt to find the positioning to maximize outs. One of the main methods worth noting is that of Murray, Ortiz, and Cho [7], in which the optimization model attempts to dually minimize the average distance to batted balls based on their "hit intensity" (a combination of launch angle and exit velocity) as well as maximize the total area of the field covered. This study uses data collected by the University of California at Santa Barbara baseball team and aims to optimally place the three outfielders using the stated technique.

The method of minimizing the average distance to the batted balls is based on the work of Swain and ReVelle [10] for which the optimization model aims to minimize the average distance between site facilities to provide service based on the magnitude of demand. This is significant for fielder optimization since proximity to a batted ball based on its hit intensity greatly influences the probability of successfully fielding the ball and generating an out.

The second methodology for which this optimization routine is based is that of Church and Revelle [4] which aims to serve all facilities with demand within some fixed maximum distance (or travel time) from the service center. Again, this is relevant to fielder optimization since fielders should be able to reach as many fly balls as possible before they touch the ground to generate the maximum number of outs and impede base runners' progress.

These two methods are combined into a multi-objective spatial optimization technique in which the outfielders are first dispersed to maximize the total weighted demand of batted balls covered given each outfielder has a fixed range. They are then re-shifted to minimize the average weighted distance to the batted balls based on the batted ball intensity. The aim is to find the Pareto optimal solution between the maximal coverage and minimal weighted distance that is conducive to generating the most outs.

One of the main advantages of this method is that the balance between the two optimizations can be tuned to favor either objective as another solution can be chosen from the Pareto front. This is practically advantageous for the staff of a baseball team since it provides multiple mathematically justified layouts that can be chosen based on the defensive philosophy of the team. This is beneficial when optimizing for a different field shape, as shape irregularities may require different defensive approaches.

This leads us to the main limitation of this method: its generality when considering the infielder configuration. The article mentions that a 3-outfielder model is not necessarily the best choice in certain scenarios since an additional outfielder can help increase coverage in a situation such as “no doubles defense”. With the rising popularity of extreme infield shifts, it is becoming ever more common to have a “hybrid” infielder who plays a shallow outfield position to the side of the field where the batter is most likely to hit the ball. Essentially, this method does not rigorously consider shifts where one or more infielders move to an outfield position and does not offer positioning insights for infielders.

This is also limited by the assumption that all fielders placed in the outfield have a constant fielding radius of 90 feet in every direction. While this is a fair, data-driven estimation of outfielder range, it does not account for decreased coverage when the fielder is closer to the infield as one would typically have less time to react to a batted ball with high exit velocity. This could artificially inflate the field coverage objective and create fielder alignments where outfielders are placed too close to the infield and miss potential outs hit behind them.

We initially used this method to build our optimization but opted for a different direction due to practical issues. Further details will be explained in Section 4.1.

2.1.3 The Shift Tester

Grove's method [5] goes beyond the previous approach by incorporating all free-moving fielders into the shift instead of outfielders exclusively. This is accomplished by adjusting the attributes of a batter (mean exit velocity, mean launch angle, flyball pull percentage, etc.) and attributes of each particular fielder (fielding range, arm strength, etc.) to create a defensive alignment that alters the positioning based on the batter's and each defender's skill.

The method aims to optimize the predicted batting average on balls in play (BABIP) and is built using various measurements from Baseball Savant [1] data to estimate the probability of an out for each batted ball. Outfielder positioning is primarily determined by the catch probability of a flyball, which considers factors such as hang time, the distance a fielder must travel to catch the ball, and outfielder range. It is also contingent on batter metrics such as average exit velocity and flyball pull percentage. A greater exit velocity typically moves the fielders away from home plate, and a greater pull percentage typically moves fielders toward the pull side of the batter. Additionally, the model (for outfielders) incorporates groundball pull percentage, although its impact is less apparent in comparison to exit velocity, launch angle, and flyball pull percentage.

The infield alignment is determined by the same variables as the outfield plus additional factors such as transfer time and arm strength. As expected, groundball pull tendency and average exit velocity influence the infield alignment the most. The shift caused by a higher groundball pull tendency tends to shift fielders toward the pull side (which coincides with conventional baseball wisdom) while higher exit velocity tends to move the fielders further from home plate, similar to that of the outfielders. Positioning fielders farther from home plate for a batter with a higher average exit velocity aligns with conventional baseball wisdom, providing infielders more reaction time to handle hard-hit balls directed towards them.

The most noteworthy advantage of this method is its ability to adjust the defensive alignment based on the skills of the individual fielders. This feature is practically advantageous as it allows a team to test different combinations of fielders and choose the fielder combination (and alignment) for which BABIP is minimized. This expedites the process of testing current players at different positions or assessing how a new player (one who has not previously played on this team) would fit into a team's defensive scheme based on known defensive metrics.

Similar advantages are derived from the adaptability of the model to the tendencies of the batter. Incorporating this makes for a whole-field defensive alignment that is tailored to each batter's strengths and weaknesses. In turn, this will allow the defense to exploit each batter's tendencies and lower the expected BABIP to generate more outs. This is not necessarily unique to this method, but it is worth mentioning since it provides batter-specific insight for each matchup.

While this model comprehensively considers batter and fielder tendencies, it does not include information on a batter's performance against a specific pitcher, which is vital in game situations where batted ball outcomes depend on the attributes of the pitcher. Consider right-handed batter A who adheres to the average metrics for right-handed batters. The spray chart for this batter will vary when facing a pitcher with an average fastball velocity of 88 mph compared to facing a pitcher with an average fastball velocity of 100 mph. It is probable that the batter pulls the ball, generates fly balls, and generates a different exit velocity (among other metrics) at a different rate between these pitchers, meaning a fielder alignment based on the batter versus an aggregate pitcher cannot accurately describe the case-by-case tendencies that this method attempts to achieve. In sum, incorporating pitching metrics into this model would increase its comprehensiveness and likely lead to an even lower estimated BABIP for an already exhaustive model.

2.1.4 Swing Shift: A Mathematical Approach to Defensive Positioning in Baseball

Similar to the method of Murray, Ortiz, and Cho [7], the method developed by Bouzarth et al. [3] utilizes an integer linear program to optimize the fielder alignment. The difference is that this method optimizes the alignment for all seven fielders instead of just the three outfielders. The optimization technique aims to maximize the coverage over the distribution of batted ball intensities, where intensity is a normalized frequency with which the batter hits the ball to a given location on the field. The primary regions are delineated as follows: the first region extends up to 75 feet from home plate, assuming that both the pitcher and catcher can handle balls hit within this area. The second region encompasses the remaining areas of the infield and outfield. The region outside of the 75 foot radius is then partitioned into disks of 5 feet in diameter that represent the possible locations for the seven other fielders to field the ball.

Similar to the other techniques, this method uses batter-specific distributions to minimize the expected BABIP. Each fielder is assigned a radius based on their proximity to home plate, and this radius is subsequently re-scaled to an ellipse shape. The major axis of the ellipse is perpendicular to the ray originating from home plate to the fielder's position. The radius scales positively and linearly as the Euclidean distance of the fielder increases from home plate. When a fielder is 210 feet or more away from home plate, his coverage shape transforms from an eccentric ellipse to a circle. The underlying logic says that outfielders typically have more time to field a ball in every direction, whereas an infielder has more time to react moving side to side than forward and backward. A similar idea was incorporated into my optimization model with different semantics.

The distribution of batted ball intensity was not exclusively determined by hit frequency. Instead, risk scores were incorporated for batted balls to emphasize the importance of those near foul lines and beyond 325 feet from home plate by giving them increased "weight". This is because balls hit near the foul line and further than 325 feet have a greater probability of being extra-base hits and increase the probability of the defense allowing more runs. Thus, it would be advantageous to give a bias to the areas for which extra-base hits are more likely to occur.

One significant advantage of this method is its flexibility to accommodate a fourth outfielder in its configurations. As aggressive defensive shifts have become more common, teams often deploy alignments where an infielder is moved to the outfield. This is advantageous since the other methods either assume a three-outfielder alignment or set a boundary for the maximum distance that an infielder can play from home plate. This constraint limits the degree the coverage can be optimized, meaning it is beneficial to allow more freedom for fielder positioning.

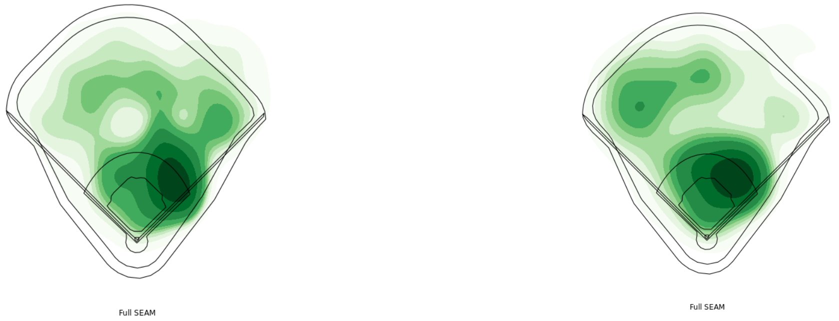
While many ad-hoc decisions were implemented to improve the practicality of the model, the most significant were statistically justified. For example, the semi-major and semi-minor axes of the infielders' fielding range were calculated by dividing the average exit velocity by the average player speed and scaled to describe the lateral and vertical coverage. It would be preferable to minimize the number of ad-hoc assumptions, but the practical nature of fielder positioning calls for some assumptions that are too complex to derive statistically, given publicly available data (such as a definition of where the outfield starts). Despite this, model testing showed a significant decrease in BABIP among nearly all tested players and confirmed the optimization process improves defensive efficiency.

That being said, the method's generality could be improved with a deeper investigation into the modeling assumptions and specifications, which are sensible but ad-hoc. A couple of these assumptions include the fielding coverage of the pitcher and catcher and where the outfield begins. It is assumed that the pitcher and catcher will field batted balls within 75 feet of home plate (so the seven other fielders will not be placed there), and the outfield begins 210 feet from home plate. The point where the outfield begins is significant since it determines the fielders' coverage shape and influences the percentage of fielded batted balls. These assumptions are logical from a traditional baseball perspective but lack statistical justification. It is advisable to determine these assumptions through statistical analysis for a more realistic and practical model.

Similarly, understanding how the risk vector adds weight to specific areas of the field and providing the rationale for these regions would be beneficial to develop a sound understanding of the model as the 325 foot cutoff and area "near" the foul line are logical from a traditional baseball perspective but are not justified otherwise. The risk vector conflicts with the paper's objective of positioning fielders where batters are most likely to hit the ball and reduce BABIP. While the risk vector was implemented to prevent extra-base hits, which aligns with decreasing SLUGIP, this complicates the optimization process's goal since all base hits (in play) have an equal BABIP weight.

While there are many minor semantic differences between this method and the method detailed in this paper, they share many of the same broad ideas. As previously stated, my approach employs a fielder range scaling function to determine the ellipse for a fielder, a concept shared by both methods. This function illustrates how fielders typically exhibit greater lateral range compared to forward and backward movement. Both methods also aim to maximize the amount of "density" covered in their respective manner. The method developed by Bouzarth et al. [3] attempts to maximize the coverage of batted ball intensity based on a normalized frequency distribution with an added risk vector, while my method attempts to maximize the estimated coverage of batted ball probability. Other similar ideas are described in more detail in their respective sections.

I attempt to build upon the work of this paper by utilizing a more sophisticated distribution as the medium of optimization. While Bouzarth et al. rely on the empirical spray chart of a batter to develop an optimal fielder alignment, this approach tailors the distribution to each individual pitcher/batter matchup. The validity of this assertion can be verified by the "Validation" section of the original SEAM article (Wapner, Dalpiaz, and Eck [11]). This is advantageous when a batter or pitcher performs differently against left or right-handed batters or pitchers, respectively. For example, Figures 2a and 2b show the SEAM distribution plots of left-handed batter Jacoby Ellsbury versus right-handed pitcher Trevor Bauer and left-handed pitcher Clayton Kershaw. As evident, there is a notable difference in the estimated outfield density between the two matchups, suggesting variations in Ellsbury's performance. Although not all-encompassing, this demonstrates how using an aggregate spray chart may cost a team valuable outs.



(a) Trevor Bauer vs Jacoby Ellsbury

(b) Clayton Kershaw vs Jacoby Ellsbury

Figure 2: RHP & LHP Disparity vs Jacoby Ellsbury

Of the mentioned criticisms, many of them have relatively simple fixes. If the method focuses on minimizing BABIP, removing the risk vector from the objective function would be trivial in practice. This would likely reduce the computational resources required for running this optimization since the computation of weights for balls hit near the foul line and deep in the outfield is excluded.

Also, finding statistical justification for the pitcher/catcher fielding region and where the outfield begins can be estimated via Baseball Savant [1] data. No other reputable sources for major league positioning data were found in my search, but the data provided via Baseball Savant remains adequate.

Despite the differences, it is relevant to mention that independently converging to a similar solution regarding the same problem is a proof of concept in itself. This validates the robustness of the methodology and underscores the reproducibility of the findings. Such convergence offers a reinforcement that the drawn conclusions are grounded in sound objective ideas and can be built upon in future work.

2.2 Optimization Basics

2.2.1 BFGS

BFGS is a popular quasi-Newton deterministic gradient descent optimization method (Nocedal and Wright [8]) used in a wide variety of scientific and engineering applications, ranging from machine learning and numerical optimization to computational chemistry and physics. The algorithm utilizes the gradient of a surface in $\mathbb{R}^n$ to guide its search for a local optimum, and it will converge to a unique solution given some initial conditions. Specifically, to minimize differentiable scalar function $f(\vec{x})$ where $\vec{x} \in \mathbb{R}^n$, we begin at initial point $\vec{x}_0$. The search direction $\vec{p}_k$ at the k^{th} step is derived by solving the following Newton-like equation:

$$B_k \vec{p}_k = -\nabla f(\vec{x}_k)$$

B_k is a symmetric positive definite quasi-Newton approximation of the Hessian matrix (a square matrix of second-order partial derivatives of f) evaluated at $\vec{x}_k$, and $\nabla f(\vec{x}_k)$ is the gradient of f evaluated at $\vec{x}_k$. Instead of calculating a new Hessian matrix at every iteration (which takes at least $\mathcal{O}(n^{2.373})$ time), B_k is updated to account for the measured curvature in the most recent step to reduce computational complexity.

Furthermore, a line search is performed to find the k^{th} step size, $\alpha_k = \text{argmin}(f(\vec{x}_k) + \alpha \vec{p}_k)$.

Following this, the solution $\vec{s}_k$ to the secant equation is updated before updating the value of $\vec{x}_k$. So $\vec{s}_k = \alpha_k \vec{p}_k$ represents the size and direction of the step, and $\vec{x}_{k+1} = \vec{x}_k + \vec{s}_k$ updates the solution vector.

Finally, to update the approximation of the Hessian matrix B_k, the vector representing the change in the gradient must be updated: $\vec{y}_k = \nabla f(\vec{x}_{k+1}) - \nabla f(\vec{x}_k)$. This is completed with the following equation:

$$B_{k+1} = B_k + \frac{\vec{y}_k \vec{y}_k^T}{\vec{y}_k^T \vec{s}_k} - \frac{B_k \vec{s}_k \vec{s}_k^T B_k}{\vec{s}_k^T B_k \vec{s}_k}$$

The stopping criterion for this method is reached when the norm of the gradient becomes less than a predetermined small value, signaling that the algorithm cannot further optimize with the current step size. Rigorously, when $\|\nabla f(\vec{x}_k)\| < \epsilon$ for fixed $\epsilon \in \mathbb{R}^+$.

BFGS is useful for several reasons in the context of this problem. First, it demonstrates computational efficiency, achieving a time complexity of $\mathcal{O}(n^2)$ compared to other Newton methods that require $\mathcal{O}(n^3)$ time due to their need for Hessian matrix inversion. This is advantageous in practice as users receive near-immediate results and can quickly render multiple alignments.

Another advantage is its large number of R package implementations in R packages. This makes it straightforward to find a robust implementation of the algorithm; the base R **optim** package sufficed for this use case.

One of the main disadvantages of BFGS (in this problem's context) is that it is a local search method, meaning it does not necessarily find the most optimal solution given the initial conditions. This is problematic for dual technique as finding solutions on the Pareto front is inhibited by a fielder getting "stuck" at local maxima, preventing a more optimal alignment.

2.2.2 Simulated annealing

Simulated annealing (SANN), unlike BFGS, is a stochastic global optimization search algorithm commonly used for large-scale discrete problems with multiple constraints. It finds applications in various fields such as solving the traveling salesman problem and predicting protein structures. SANN prioritizes approximating the global optimum over swiftly finding a local optimum, distinguishing it from BFGS.

The simulated annealing algorithm gradually reduces the likelihood of accepting suboptimal solutions while randomly exploring the solution space. At each step, the algorithm randomly selects a neighboring solution and compares its quality with the current solution. If the neighboring solution offers a better evaluation of the objective function, the algorithm moves to this new location. The temperature parameter controls the acceptance probability, meaning a lower temperature will search a more focused space. As the temperature parameter decreases to zero, the algorithm settles on an approximation of the global optimal solution.

Algorithm 1 Simulated_Annealing

```
$x \leftarrow x_0$                                                    ▷ starting state

for $k = 0 \to k_{\max} - 1$ do          ▷ iterate through steps that calculate temperature
    $t \leftarrow$ temperature$(1 - \frac{k+1}{k_{\max}})$                      ▷ setting the temperature
    $x_{\text{random}} \leftarrow$ neighbor$(x)$                      ▷ randomly choosing a neighbor of $x$
    if $P(E(x), E(x_{\text{random}}), t) \geq$ random$(0, 1)$ then    ▷ determine acceptance probability
        $x \leftarrow x_{\text{random}}$
    end if
end for

return $x$
```

P calculates the acceptance probability based on the energy of the states and current temperature. I will omit the semantics, but the acceptance probability decreases as the temperature and $E(x_{\text{random}}) - E(x)$ increase. States with low energy are considered more optimal. The energy function E is analogous to the objective function which evaluates the x location independently of temperature. The goal of this algorithm is to bring the system to a state with the minimum possible energy.

This method is notable for its ability to conduct a comprehensive search to find an approximate global solution. Given the aim is to find the global optimal fielder alignment to minimize predicted BABIP, this method is particularly useful when evaluating under a single objective. By heuristic observation, simulated annealing tends to find fielder alignments that cover more high-density areas of the distribution than BFGS. In practice, the difference between the BFGS-generated and simulated annealing-generated alignments would likely result in a couple of outs per game. Thus, in a high-stakes scenario such as an MLB game, a user would likely prefer simulated annealing as it typically leads to a more optimal solution.

The main disadvantage of simulated annealing is its computational complexity. While simulated annealing produces more optimal alignments in multiple minutes, BFGS renders solutions in a matter of seconds. This was anticipated based on each algorithm definition but posed an obstacle when testing multiple fielder combinations sequentially. This could pose practical challenges for users with less powerful computing resources who need to generate numerous fielder alignments, a common task for MLB teams due to the multitude of possible batter-pitcher matchups in each game.

3 Data

3.1 SEAM Data

The data detailing the pitcher-batter distributions is taken from the SEAM Github[1] which acts as a proprietary package that allows for data and function calls from SEAM's framework. The SEAM distributions are calculated using Baseball Savant [1] data taken from 2017 to 2021, which includes every MLB pitcher vs. batter event in the said time frame. Data was collected via the TrackMan system from 2017 to 2019 and in 2020 transitioned to the Hawk-Eye tracking system for even more detailed measurements.

The SEAM distribution is a square matrix $D \in \mathcal{M}_{100\times 100}(\mathbb{R})$, where each entry $d_{(i,j)} \in D$ represents the probability that the batter will hit the ball to the (i, j) coordinate. To extract the shape of each major league field from the square matrix, a data filtering process rescales and filters the matrix coordinates to fit within fair territory. Since every SEAM distribution has the same coordinates, the required stadium coordinates were only calculated once. Then for every generated SEAM distribution, the SEAM coordinates would be matched to those of the desired field to produce a park-specific surface.

The choice to optimize over the in-play field shape was made since the objective of this work is to find an optimal pre-pitch positioning to minimize BABIP. Fielders cannot stand outside of fair territory during a pitch, meaning the only possible choices are exactly within the in-play field cutout. This raises the question of whether it would be more effective to place fielders near the foul line given a batter hits many foul balls. While this is a valid observation, the scope of this work is to study balls in play. For the same reason, it is unreasonable to move fielders backward given the batter hits many home runs since home runs are not playable.

A copy of the cutout where the z values are constant is linearly interpolated to generate a denser distribution for the area maximization portion of the dual technique to run on. This created a $\mathcal{M}_{n\times n}(1)$ matrix for each of the stadiums where $246 \leq n \leq 262$, stadium dependent. This was implemented due to practical issues with a BFGS method on the original cutout; further explanation can be found in Section 2.2.1.

[1] SEAM Github

3.2 Positioning Data

Before beginning the optimization techniques, statistically justified initial coordinates for each fielder were required. These were found by referencing the Baseball Savant [1] average fielder positioning data to set the initial alignment for right-handed and left-handed batters. The averages were taken from the 2022 season. The difference between the average fielder positions over the past five years is negligible, so it was assumed the 2022 average placements would be adequate.

3.3 Baserunning Data

Furthermore, all batted-ball Baseball Savant data from 2017 to 2022 was used to fit a batter-specific model to predict launch angle and exit velocity at each field coordinate. This was used to predict the slugging percentage for each point on the field to incorporate the base-running ability of the player since different players will generate varying outcomes for the same hit (e.g. a ball hit down the line might be a triple instead of a double for fast players). More details can be found in Section 7.

I will now discuss the optimization techniques.

4 Optimization Techniques

4.1 Dual Technique

The initial approach to the problem, as proposed by Murray, Ortiz, and Cho [7], involves a two-step process. The technique begins by maximizing the total weighted batted balls covered by the fielders. Subsequently, a minimization step is applied to reduce the average weighted distance to the points of demand. This two-step approach aims to find solutions on the Pareto front to provide the user more autonomy in choosing their positioning strategy. The constraints are adapted from Murray, Ortiz, and Cho [7] and characterize the following:

1. More than one fielder cannot be assigned to one location.

2. There are 7 free-moving fielders.

Before expressing these constraints formally, I will define the following:

- a_i = the batted ball intensity at position i, where i is a coordinate tuple
- d_{ij} = the Euclidean distance from position i to j
- X_{ij} = an indicator of whether a player at position j is assigned to field a batted ball at location i
- i = the index of the field grid for batted ball locations
- j = the index of the field grid for fielder positions

In the context of the SEAM field cutout, the set of i locations is equal to the set of j locations since the batted ball locations are the same as the potential fielder positions.

The dual optimization process is defined as follows:

$$\text{maximize} \quad \sum_i \sum_j a_i X_{ij} \tag{1}$$

$$\text{minimize} \quad \sum_i \sum_j a_i d_{ij} X_{ij} \tag{2}$$

$$\text{subject to} \quad \sum_j X_{ij} = 1 \tag{3}$$

$$\sum_i X_{ij} = 7 \tag{4}$$

This dual-objective optimization aims to generate a Pareto front, allowing the user to select an alignment that agrees with their strategy without compromising optimality. The Pareto front is comprised of optimal alignments where increasing maximized coverage comes at the expense of minimized weighted distance, and vice versa. This freedom empowers users to choose an alignment that harmonizes fielding philosophies and identifies the optimal defensive configuration.

The two main gradient scaling techniques used were the Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) and simulated annealing; both are iterative methods for solving nonlinear optimization problems. One may reference Sections 2.2.1 and 2.2.2 for technical background.

As stated, the dual method iterates over each fielder and optimizes their position to maximize the total sum of the weighted batted balls covered. When employing BFGS with the base R **optim** function, a challenge arose where no fielders would shift from their initial positions. The problem stemmed from limitations on step size: if too small, fielders would be stuck in their starting positions due to a lack of nearby points, and if too large, they would end up in impractical positions. To address this, a field distribution with linearly interpolated coordinates was created. This increased the plot's density while preserving the integrity of the original distribution. These interpolated distributions are about six times more dense than the original. The result is a finer grid that allows for spatial maximization with no issues regarding its step size.

After obtaining the fielder alignment that maximizes coverage, this is used as the initial condition for the distance minimization subroutine of the dual technique. This routine uses simulated annealing with a limit on the number of iterations to prevent the minimization from converging to a global optimum as the objective is to find a Pareto optimal solution. The starting temperature, number of iterations spent optimizing each player's positioning, and number of iterations over all seven fielders were derived through experimentation. I will now detail the implementation of the iterative optimization method used in this and all subsequent described optimization techniques.

Suppose from the set P of seven players, I choose player $i_1 \in P$ to be the first player to move. I iterate through $P \setminus \{i_1\}$ and calculate each player's fielding coverage based on their location and distance from home plate. The portion of the distribution that each $j \in P \setminus \{i_1\}$ covers is removed from the original distribution. This leaves player i_1 with a distribution for which he will not grossly overlap coverage with other players after optimization; this also prevents double-counting of coordinates that multiple fielders can reach.

Optimization technique m (either BFGS or simulated annealing) then runs on player i_1 and optimizes their position over the objective function. Once it converges, the new coordinates are saved and the temporary distribution is reset to the original. The algorithm iterates to the next fielder. The algorithm repeats this process for each fielder for a user-fixed number of iterations.

Algorithm 2 Iterative_Optimizer

```
for $k = 1 \to N$ do                              ▷ number of iterations over all players
    for $i \in P$ do                              ▷ player whose position is optimized
        $D \leftarrow S$                          ▷ setting temporary distribution
        for $j \in P \setminus \{i\}$ do
            $R \leftarrow Find\_Region(j)$        ▷ finding player coverage with coordinates
            $D \leftarrow D \setminus R$          ▷ cutting coverage from field
        end for
        $i \leftarrow Optimize(i, D, m)$          ▷ optimizing $i$'s position over distribution $D$ via $m$
    end for
end for

return $P$                                        ▷ returning the optimized coordinates
```

The *Find_Region* helper function described in Section 4.1.1 finds a fielder's coverage.

This method works relatively well for finding fielder alignments, but various technical issues prevent this method from being as analytically sound as others.

The first main issue of the dual method is the difficulty of measuring the Pareto front using gradient descent. In contrast to the method described by Murray, Ortiz, and Cho [7], the distribution optimized over is relatively dense and requires a method that can efficiently walk over it. It is also relevant to mention that given the SEAM surface, the problem's complexity explodes if written as a linear program. The original implementation is a dual-objective linear program that can be solved relatively efficiently due to data sparsity, as it uses 85 balls in play in opposition to the several thousand in this application. Due to the density, it is more reasonable to use gradient descent.

The second main issue is the difficulty of measuring the Pareto front using gradient descent. As mentioned, evaluating, optimizing, and comparing solutions over a dense distribution is resource-intensive and time-consuming, making it arduous to evaluate and store a large number of these candidate solutions. Thus, the calculated "optimal" solutions via gradient descent may not be optimal as better alignments may not have been explored over the grid.

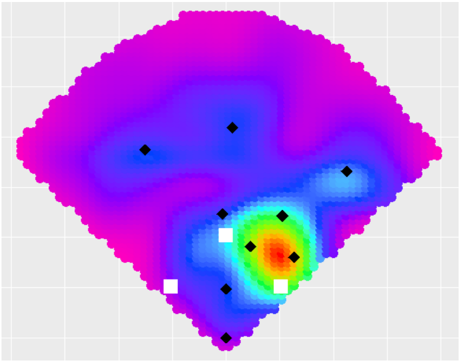


Figure 3: Impractical Alignment

Because of this, "optimal" fielder alignments may appear nonoptimal or impractical. While this is not a mathematically justified way of measuring a solution, part of the objective of this work is to provide a tool for which a user can find a fielder alignment to assist their defensive strategy. If an alignment is mathematically sound but is not practical in the traditional baseball sense, the alignment will not be used. For example, reference Figure 3. While the fielders (black diamonds) seem to cover the areas where a ball is likely to be hit, placing a fielder behind second base instead of on the third base side of the infield would be impractical as a batter could simply bunt to the third base side to achieve a hit. Thus, if this were recommended to an MLB manager, this criticism would outweigh the mathematical correctness that the optimization provides.

Due to these issues, I moved forward with other optimization techniques to find more practical, but still mathematically justified, fielder alignments. I will now define the *Find_Region* function and continue by describing these alternative techniques.

4.1.1 Find Region

To optimize each fielder's position relative to the other players, the range of each player must be determined. To begin, it is necessary to accurately scale the players' radii. This involved selecting three distances from home plate based on aggregate fielder positioning data from Baseball Savant [1]. The chosen distances of 0, 140, and 250 feet from home plate were fit with a quadratic regression model.

To model the fielding radius based on distance from home plate, the function f is defined to map a fielder's Euclidean distance from home plate to some positive radius; $f : \mathbb{R} \to \mathbb{R}$. It was assumed that a player standing at home plate (like a catcher) has a fielding radius of 10 feet, which is the typical distance of a successfully fielded bunt. The other two radii were approximated through experimentation as fielder range data from MLB games is not readily available to the public. The three function estimates used to fit the radius function are the following: $f(0) = 10$, $f(140) = 30$, and $f(250) = 50$. These are measured in feet.

The equation of the regression model is approximately as follows: $f(x) = 0.0002x^2 + 0.121x + 10$. A piecewise element was added for which the fielder range is limited to 50 feet when a fielder is more than 250 feet from home plate. This is meant to represent the physical limitations of an outfielder as a player's range is limited by their sprint speed. Therefore, the function to calculate the fielder's range is:

$$f(x) = \begin{cases} 0.0002x^2 + 0.121x + 10, \; 0 \leq x \leq 250 \\ 50, \; x > 250 \end{cases} \tag{5}$$

Since a fielder has more time to react to a ball hit to their left or right rather than directly toward them, the semi-minor axis was scaled by 0.85 for radii in the $0 \leq x \leq 250$ range.

As described, each player has an elliptical fielding range with the major axis parallel to the x-axis of the field. This does not fully encapsulate the fielding range of the player since fielders typically orient their body toward the batter. Thus, a rotation is applied to each of the ellipses before proceeding.

Suppose the coordinates of player j are $(x_j, y_j) \in \mathbb{R}^2$. First, the angle θ with respect to the axis beginning at home plate in the direction of the pitcher's mound is found by calculating $\theta = -\arctan \frac{y_j}{x_j} - \frac{\pi}{2}$. The extra $\frac{\pi}{2}$ term sets the original positive y-axis to the zero rotation. This was done to simplify computations. Calculating the fielder's ellipse before rotating would cause issues defining which coordinates are inside of the ellipse after rotation. To work around this, the whole distribution is rotated instead and the respective ellipse is calculated on the rotated distribution. The coordinate system uses a simple 2-dimensional forward rotation formula:

$$\begin{cases} x_{\text{new}} = x_{\text{original}} \cos(\theta) - y_{\text{original}} \sin(\theta) \\ y_{\text{new}} = y_{\text{original}} \cos(\theta) + x_{\text{original}} \sin(\theta) \end{cases}$$

After the ellipse is calculated, the coordinate system is re-rotated by $-\theta$ to the original coordinate system for the next iteration.

$$\begin{cases} x_{\text{original}} = x_{\text{new}} \cos(-\theta) - y_{\text{new}} \sin(-\theta) \\ y_{\text{original}} = y_{\text{new}} \cos(-\theta) + x_{\text{new}} \sin(-\theta) \end{cases}$$

The *Find_Region* algorithm to find a fielder's coverage is defined as follows:

Algorithm 3 Find_Region

```
$(x, y) \leftarrow (x_0, y_0)$    ▷ player coordinates
$\theta \leftarrow -\arctan(\frac{y}{x}) - \frac{\pi}{2}$
$D \leftarrow S$    ▷ set of distribution coordinates $(d_x, d_y) \in D$

$(x_{\text{range}}, y_{\text{range}}) \leftarrow f(\sqrt{x^2 + y^2})$    ▷ fielder range defined by 5
$(x_{\text{rotated}}, y_{\text{rotated}}) \leftarrow (x\cos\theta - y\sin\theta, y\cos\theta + x\sin\theta)$    ▷ rotating player coordinates

for $(d_x, d_y) \in D$ do    ▷ iterating through distribution coordinates
    $(d_x, d_y) \leftarrow (d_x\cos\theta - d_y\sin\theta, d_y\cos\theta + d_x\sin\theta)$    ▷ rotating coordinates
    if $\sqrt{\frac{(d_x - x_{\text{rotated}})^2}{x^2_{\text{range}}} + \frac{(d_y - y_{\text{rotated}})^2}{y^2_{\text{range}}}} > 1$ then    ▷ checking if coords are in fielder ellipse
        $D \leftarrow D \setminus (d_x, d_y)$    ▷ removing those not in fielder's ellipse
    end if
    $(d_x, d_y) \leftarrow (d_x\cos(-\theta) - d_y\sin(-\theta), d_y\cos(-\theta) + d_x\sin(-\theta))$    ▷ re-rotating
end for

return $D$    ▷ returning set of coordinates within a fielder's ellipse
```

4.2 Distance Minimization

After attempting the dual method and running into various issues, I tested the distance minimization subroutine itself. This technique aims to minimize the distance of each player to the coordinates of the distribution with high intensity. Informally, the batted ball intensity is multiplied by the Euclidean distance to the player for every point in the distribution. This is the objective function to minimize this average weighted distance to areas of the greatest demand.

This is described formally below.

Algorithm 4 Minimized_Weighted_Distance

```
$w \leftarrow [\,] * |D|$    ▷ creating empty list of length $|D|$ to store weights

for $i \leftarrow 1$ to $|D|$ do    ▷ iterating through distribution coordinates
    $(d_x, d_y, d_z) \leftarrow D[i]$
    $min\_distance \leftarrow \infty$

    for $j \leftarrow 1$ to $|P|$ do    ▷ iterating through the set of fielders $P$
        $(p_x, p_y) \leftarrow P[j]$    ▷ extracting fielder coordinates
        $min\_distance = \min\{min\_distance, \sqrt{(p_x - d_x)^2 + (p_y - d_y)^2}\}$
    end for
    $w[i] \leftarrow min\_distance * d_z$    ▷ store weighted distance
end for

return $\sum_k w[k]$    ▷ return the weighted sum
```

Although this implementation was sound and provided fielder alignments, the alignments exhibited similar irregularities to those described earlier that render the results impractical for use. Concurrently tested with the density maximization method described in Section 4.3, the maximization method yielded more practical outcomes. Thus, I moved forward with the density maximization method instead.

4.3 Density Maximization

Similar to the distance minimization technique, the "density maximization" technique builds off of a dual method subroutine. The goal is for each player to cover the greatest amount of density within their fielding radius. The semantics of the optimization implementation are the same as the Dual method as I use Algorithm 2 to optimize each fielder's position iteratively. The objective function for the density maximization method is detailed by Algorithm 5 below.

Algorithm 5 Density_Maximization

```
$F \leftarrow Find\_Region(d_x, d_y)$        ▷ finding the region of covered player density
$fielder\_sum = sum(F.z)$        ▷ density sum of $z$ over covered region

if $player\_position$ = "first base" then
    $distance\_from\_first = \sqrt{(p_3^x - 90/\sqrt{2})^2 + (p_3^y - 90/\sqrt{2})^2}$

    if $distance\_from\_first > 33.76$ then        ▷ checking first baseman distance
        return $-\infty$        ▷ barrier function
    end if
end if

return $fielder\_sum$
```

The implementation of the *Density_Maximization* algorithm is relatively simple as it sums the density found by the *Find_Region* function and applies a barrier to the first baseman. The barrier function checks whether the first baseman is within 33.76 feet of first base as he must be in the base's vicinity to field a throw from another fielder. This is relevant in a baseball context as nearly all fielded ground balls with no baserunners result in a throw to first base. The maximum distance of 33.76 feet was estimated using the positioning data mentioned in Section 3.2.

The barrier function is the only constraint in the final implementation as tuning the model allowed the deletion of other constraints; other constraints may be added to reflect the recent MLB rule changes restricting fielder shifts. The following are among the tested constraints: a density cutout of the pitcher and catcher fielding radii to represent their fielding ranges, a minimum distance an infielder must be placed from home plate to prevent an unreasonable alignment (infielders typically do not play close to home in standard scenarios), and that infielders must be placed outside of the base paths (to prevent the same issue). These constraints are worth mentioning as they progressed implementation; the technical details are omitted for brevity.

This continuous nonlinear optimization process can be written formally:

$$\text{maximize} \quad \sum_{p \in P} \sum_{d \in D} a_d X_{dp} + \mathbb{1}_{\left\{\sqrt{(p_3^x - 90/\sqrt{2})^2 + (p_3^y - 90/\sqrt{2})^2} > 33.76\right\}}$$

(p_3^x, p_3^y) are the (x, y) coordinates of the first baseman; the indicator represents his barrier function and is defined as follows:

$$\mathbb{1}_{\{\cdot\}} = \begin{cases} -\infty, & \sqrt{(p_3^x - 90/\sqrt{2})^2 + (p_3^y - 90/\sqrt{2})^2} > 33.76 \\ 0, & \text{else} \end{cases}$$

The objective function attempts to maximize the weighted sum of the density $d \in D$ that each player $p \in P$ can reach. The points that each player can reach are described by the *Find_Region* Algorithm 3. The algorithmic outline of this optimization method is described by the *Iterative_Optimizer* Algorithm 2.

The optimization technique used for density maximization is simulated annealing. One may find the justification for this over BFGS in Section 4.1.

I will now discuss the two main packages used and justify why the **GenSA** package (Xiang et al. [13]) is preferred to **optim** of the **stats** package (R Core Team [9]) for simulated annealing.

4.3.1 R Optimization Packages

optim is a general-purpose optimization method that is capable of various optimization techniques from quasi-Newton methods to conjugate-gradient algorithms to stochastic optimizers such as simulated annealing. This package worked well for the exploratory implementation, but I discovered that the **GenSA** package allows for finer control over simulated annealing's parameters and faster performance in comparison.

Specifically, the **optim** function allows the user to set only the maximum number of iterations and the number of function evaluations at each temperature. In contrast, **GenSA** allows the user to control both of these parameters among others such as acceptance parameters, maximum run time, improvement stopping constraints, and surface type.

GenSA works relatively well in application, but various minor issues arose due to its implementation. Repeated runs on the same distribution with the same initial conditions yielded the same optimized fielder alignment. This contradicts that simulated annealing is a stochastic technique; **GenSA** performs a random search but returns the same alignment due to the package's C++ engine seed.[2] Thus, **GenSA** can be considered random with perturbations to the initial conditions. This issue is not detrimental to the result but is worth noting as running a specific matchup multiple times (for testing or comparison pur-

[2] GenSA Parameter Seed

poses) is redundant.

Another minor issue is that **GenSA** ignores call and time limits when performing a local search to prevent the algorithm from stopping in a "search valley" and producing a local sub-optimal solution.[3] Again, this does not greatly alter performance but limits the user's control of the method. This was discovered while testing how the number of iterations decreases solution optimality; the algorithm would continually pass the number of set iterations to complete its local search. A possible remedy for this is a simulated annealing method written from scratch to allow full control over parameters. This would also allow for more troubleshooting flexibility as the base code could be altered to fit the problem's specific needs. Many simulated annealing implementations already exist, so this is a straightforward fix.

4.3.2 Density Maximization Critiques

The density maximization technique achieves its desired goal, but there are many possible improvements to bolster its mathematical and practical rigor. Of these, implementing fielder shapes with more substantial mathematical rationale is among, if not the most important. The current ellipses inadequately represent fielders' abilities as they generally demonstrate greater range when moving forward and to their glove side in reality. Thus, a more realistic fielder range would be shifted toward the player's glove side and may have a cone-like shape pointing toward home plate. The choice to use ellipses was for implementation simplicity; Section 7 gives statistical justification for these ellipses using predictive modeling on balls in play.

An initial goal of this work was to find a fielder alignment that minimizes SLUGIP as well as BABIP, so it would be desirable to incorporate the predicted locations of hard-hit balls into the distribution. Section 2.1.1 explains how the SEAM distribution describes the probability of a ball being hit to a certain location instead of accounting for base advancement. Thus, developing a density that incorporates batted-ball metrics would be advantageous; one may reference Section 7 for more on the attempts of this.

[3] GenSA Run Time

5 Visualizations

In this section, I present various fielder alignments for qualitatively and quantitatively relevant matchups to provide greater context to baseball's history in the current data-driven era and show the exploitation of batter tendencies.

While this matchup did not occur in the MLB, the faceoff between Angel's teammates Shohei Ohtani and Mike Trout for the final out of the 2023 World Baseball Classic represented a historic moment in baseball history as two of the most dominant players of the current era faced off for the first time to decide baseball's international title. Despite this matchup resulting in a strikeout to crown Japan the winner, Figure 4(a) below illustrates an optimal alignment for the event the ball was put into play.

Furthermore, Figure 4(b) shows an optimal fielder alignment for the matchup during the 2020 World Series Game 6 between Mookie Betts and Blake Snell. Despite striking out Betts in both previous at-bats, starter Blake Snell was substituted for Nick Anderson during the bottom of the sixth inning with a runner on first base. Although metrics indicate batters have a significantly greater on-base percentage in their third at-bat against a pitcher in a game,[4] Betts ultimately doubled to left field and began a 2 run rally to put the Dodgers ahead. These runs ultimately resulted in a Dodgers win that decided the 2020 World Series.

The subsequent plot in 4(c) depicts the alignment for the impossible matchup of Shohei Ohtani vs Shohei Ohtani. Two-way players have become a rarity in the modern MLB due to the elevated frequency of position specialization, and it comes even less frequent that one of these players performs at an All-Star level on both sides. While we will never witness Shohei Ohtani's prowess as a batter against his pitching counterpart, the debate over which role—batter or pitcher—holds greater dominance will forever linger among fans.

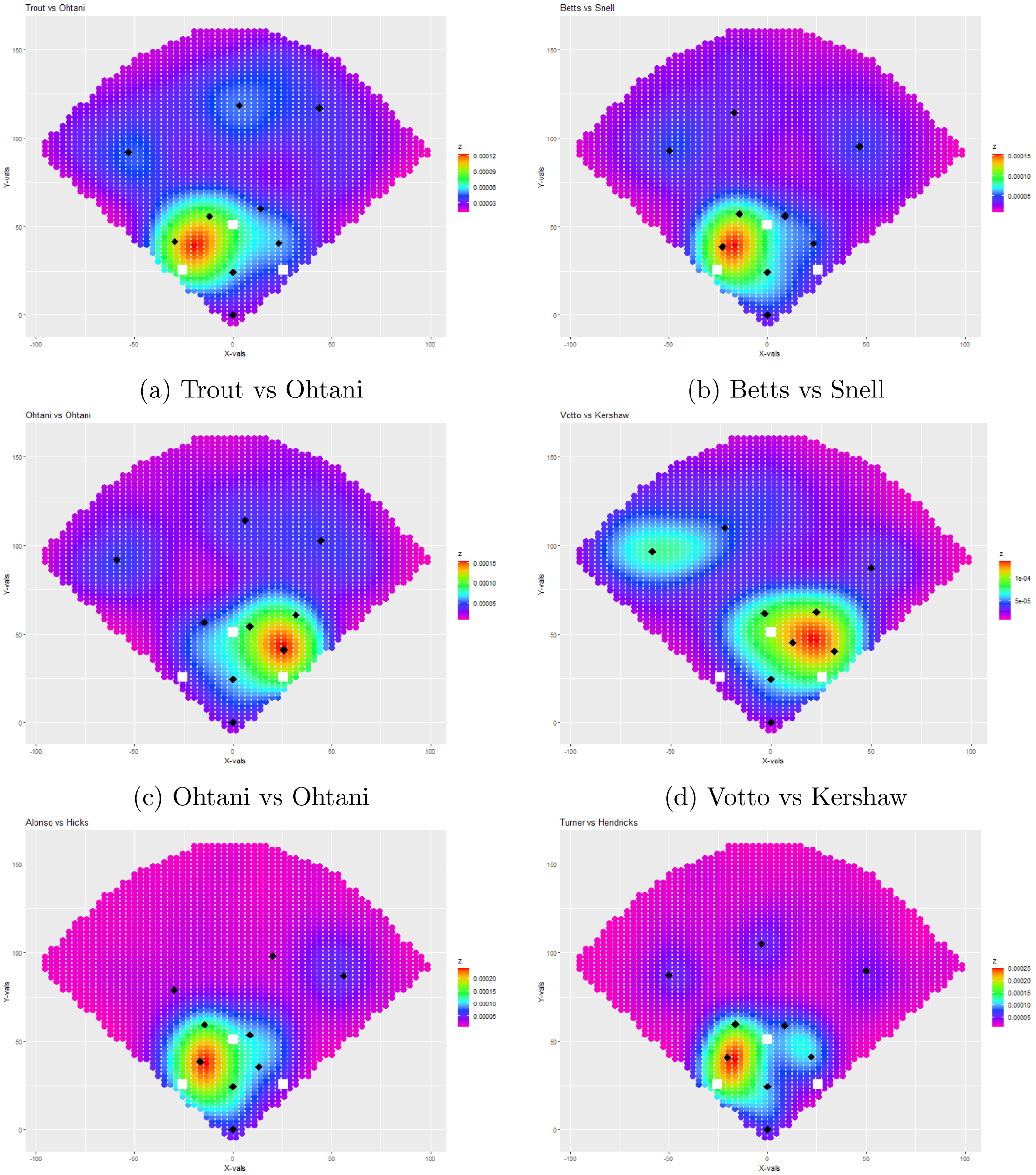
Although past their years of peak dominance, Joey Votto and Clayton Kershaw are active players who have maintained supremacy throughout the Post-Steroid and modern Statcast MLB eras. The finer statistical details and ball-flight metrics from their early matchups remained unrecorded as advanced tracking systems were not introduced to the MLB until 2015. Notably, the players began their MLB careers in 2007 and 2008, respectively. Still, the data collected from 2015 onward allows us to optimize a fielder alignment between these two future MLB Hall of Famers, as shown in Figure 4(d).

[4] Times through the order penalty.

From a quantitative lens, the plots in Figure 4 (e) and (f) show matchups between renowned power players in Pete Alonso vs Jordan Hicks, and players known for their finesse in Trea Turner vs Kyle Hendricks. These allow for comparison among varying approaches and illustrate how a defense must adjust to each strategy.

The remaining plots in Figure 4 (g), (h), (i), and (j) display matchups between batters with distinct hitting tendencies and pitchers with corresponding vulnerabilities: a batter who frequently pulls the ball against a pitcher prone to giving up contact on the pull side, a batter who consistently hits to the opposite field facing a pitcher vulnerable to opposite-side contact, a batter who pulls the ball versus a pitcher susceptible to opposite-field hits, and a batter who hits to the opposite field against a pitcher prone to pull-side contact.

5.1 Figures



(a) Trout vs Ohtani

(b) Betts vs Snell

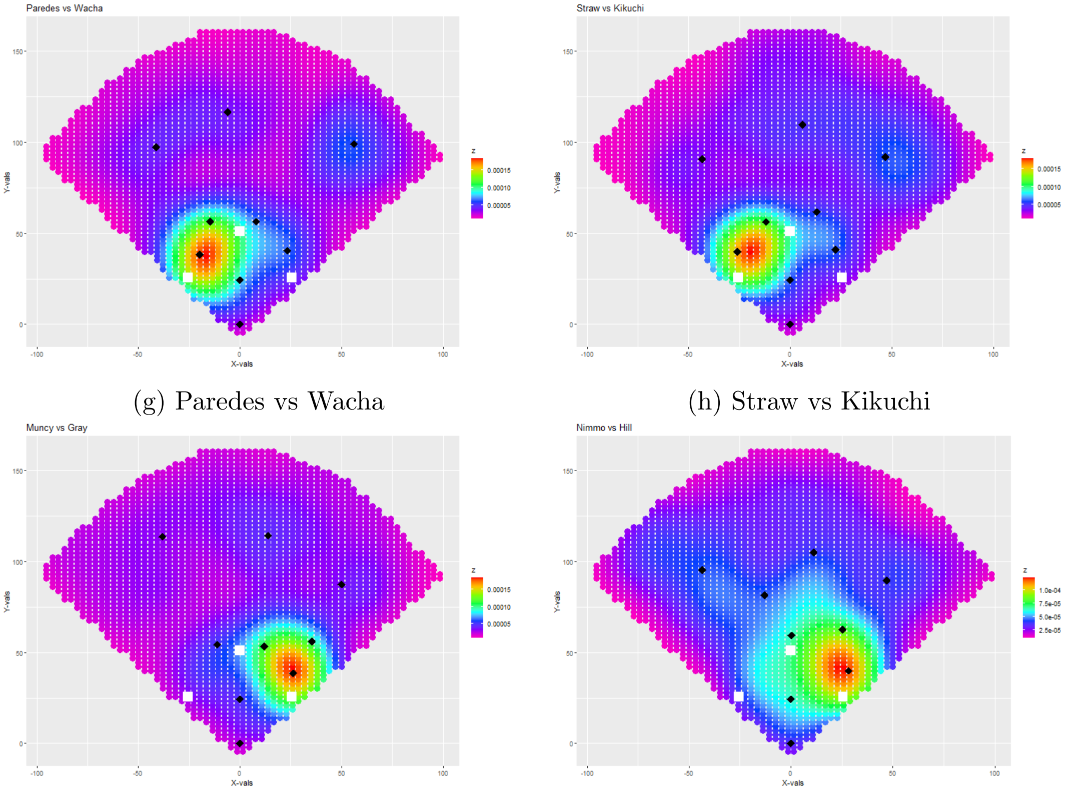
(c) Ohtani vs Ohtani

(d) Votto vs Kershaw

(e) Alonso vs Hicks

(f) Turner vs Hendricks

Figure 4: Relevant Matchups



(g) Paredes vs Wacha

(h) Straw vs Kikuchi

(i) Muncy vs Gray

(j) Nimmo vs Hill

Figure 4 (cont.): Relevant Matchups

6 Results and Validation

To demonstrate the optimization ability of the maximization method described in Section 4.3, alignments for a variety of batter-pitcher matchups were generated to verify its ability to produce a defensive advantage. Most batters used for validation in Bouzarth et al. [3] were used for direct comparison. Retired players were replaced due to their absence in the 2022 season. These players include Edwin Encarnación, José Martínez, Buster Posey, Todd Frazier, and Jay Bruce. They were replaced by randomly selecting qualified batters[5] from Statcast without replacement until the required number of right-handed and left-handed batters was satisfied. The replacement batters are Hunter Renfroe, Trey Mancini, Vladimir Guerrero Jr., Taylor Ward, and Tony Kemp.

The density maximization method requires a specific pitcher for each alignment so three qualified right-handed and left-handed pitchers were selected from the official 2022 MLB player pitching statistic leaders website.[6] For each throwing hand, the pitcher with the median of these metrics was selected: earned run average (ERA), walks and hits per inning pitched (WHIP), and the opposing team's batting average against (AVG). The list of qualified pitchers contains those who pitched at least one inning per team game during the regular season. The selected right-handed pitchers are the following:

$$\begin{cases}\text{ERA median: Logan Gilbert}\\ \text{WHIP median: Merrill Kelly}\\ \text{AVG median: Kyle Wright}\end{cases}$$

The selected left-handed pitchers are the following:

$$\begin{cases}\text{ERA median: Jordan Montgomery}\\ \text{WHIP median: Framber Valdez}\\ \text{AVG median: Robbie Ray}\end{cases}$$

The first validation metric BABIP utilized 10,000 batted balls sampled with replacement from each batter's 2022 spray chart of balls in play against all pitchers. A ball within a fielder's radius was considered an out, while a ball outside of these radii was considered a hit.

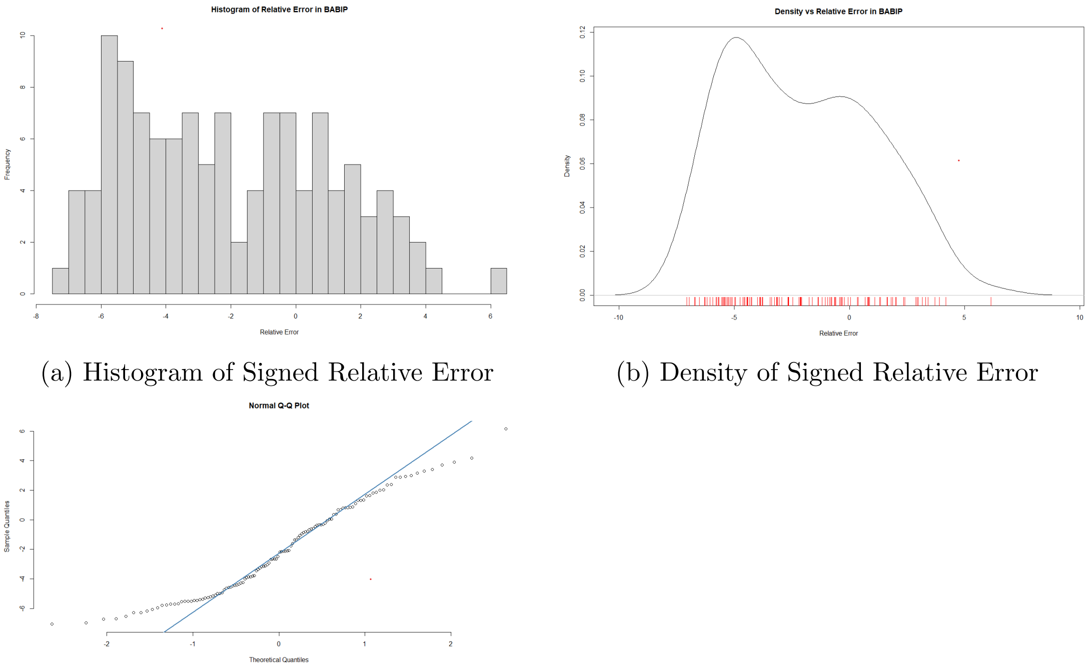
The median percent reduction in BABIP in terms of the signed relative error was approximately −2.33%; the corresponding mean was approximately −2.05%. Table 4 in the

[5] Batters with at least 2.1 plate appearances per game in 2022.

[6] MLB Pitching Statistics Leaders

Appendix B shows the percent change in BABIP for each of the tested matchups.[7]

The following plots describe the distribution of percent change in BABIP across all tested matchups.



(a) Histogram of Signed Relative Error

(b) Density of Signed Relative Error

(c) QQ-Plot

Figure 5: BABIP Distribution Analysis

The histogram and density plot of the percent signed relative error have a near-bimodal distribution; the QQ-plot confirms this. The Shapiro-Wilk normality test yields a p-value of $6.664 * 10^{-4}$, meaning I reject the null hypothesis that the data is normally distributed. An asymptotic one-sample two-sided Kolmogorov-Smirnov test yields a p-value of $< 2.2 * 10^{-16}$; I reject the null and conclude that the data does not originate from a normal distribution.

The following are the deciles of the percent change in BABIP.

[7] Negative percentages indicate that BABIP decreased after optimization.

Table 1: Deciles of % BABIP Change

	Percentile	BABIP % Change
1	0%	-7.058
2	10%	-5.700
3	20%	-5.263
4	30%	-4.438
5	40%	-3.404
6	50%	-2.331
7	60%	-1.014
8	70%	-0.279
9	80%	0.848
10	90%	2.357
11	100%	6.143

The analysis of matchups between various handedness of the batters and pitchers (RHP vs LHB, LHP vs LHB, RHP vs RHB, and RHP vs LHB) showed that the bimodality of the aggregate plot is a result of the right-handed batters having a median BABIP reduction centered near -5% while that of the left-handed batters is near -1%. The QQ-plots for BABIP reduction show a right skew for right-handed batters and near symmetry for left-handed batters.

The following are the best and worst five of all tested validation matchups by predicted BABIP reduction, respectively.

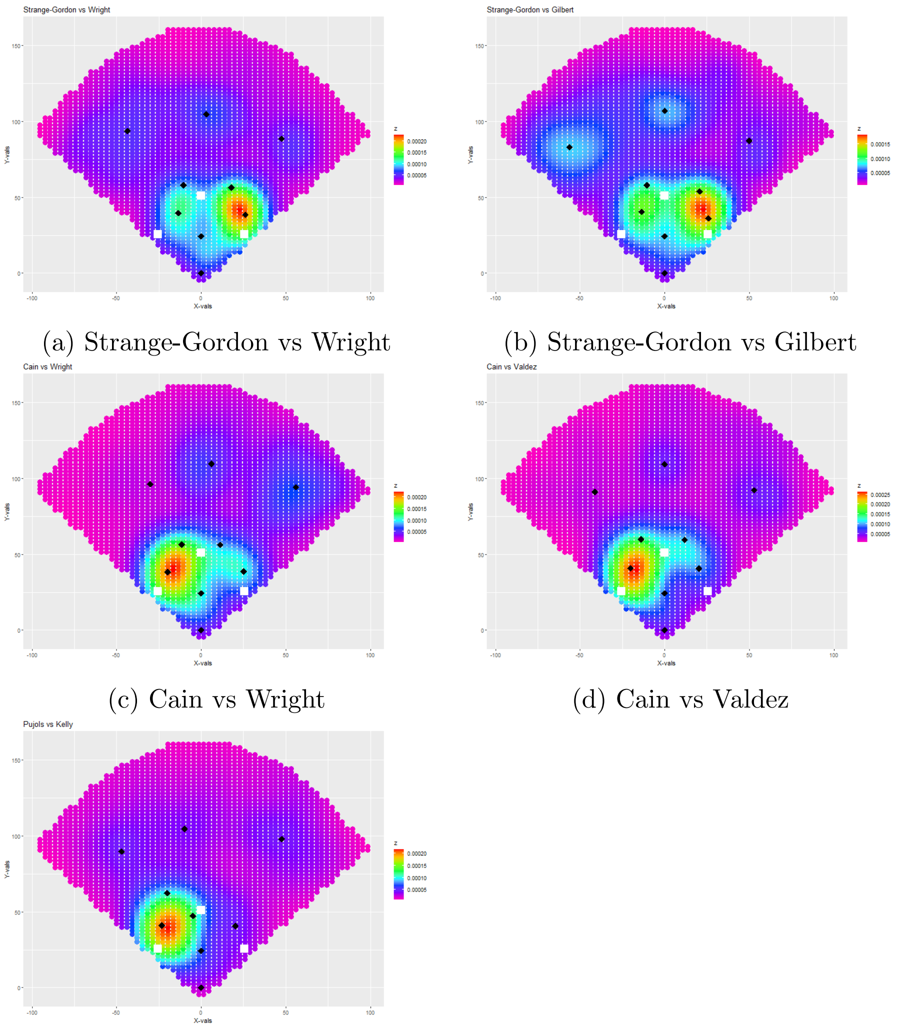
Table 2: Best Five Alignments

	Batter	Pitcher	BABIP % Change
1	Dee Strange-Gordon	Kyle Wright	-7.058
2	Dee Strange-Gordon	Logan Gilbert	-6.956
3	Lorenzo Cain	Kyle Wright	-6.724
4	Lorenzo Cain	Framber Valdez	-6.695
5	Albert Pujols	Merrill Kelly	-6.520

Table 3: Worst Five Alignments

	Batter	**Pitcher**	**BABIP % Change**
1	Bryce Harper	Jordan Montgomery	6.143
2	Freddie Freeman	Jordan Montgomery	4.182
3	Anthony Rizzo	Framber Valdez	3.892
4	Joey Gallo	Framber Valdez	3.703
5	Bryce Harper	Framber Valdez	3.413

Figure 6 below shows the plots of the best five alignments given in Table 2.



(a) Strange-Gordon vs Wright

(b) Strange-Gordon vs Gilbert

(c) Cain vs Wright

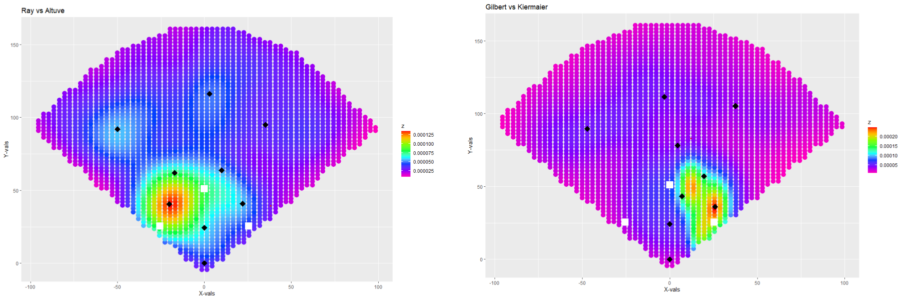
(d) Cain vs Valdez

(e) Pujols vs Kelly

Figure 6: Plots of the Best Five Alignments

Upon inspecting the raw data, seven left-handed batters yielded an increase in predicted BABIP for at least half of their tested matchups. I hypothesized that a low similarity score between a batter and pitcher results in a low BABIP error reduction since the training distribution (SEAM) lacks matchup specificity. A linear model predicting the BABIP error reduction using the log transformation of the similarity score (derived from SEAM) as the independent variable yielded a low negative correlation. Explicitly, an increase in log similarity score decreases the predicted BABIP; this implies that better training data loosely leads to more optimal alignments.

Although the weak correlation indicates that the similarity score cannot fully explain BABIP error, it is relevant to show cases for which the SEAM distribution greatly influenced BABIP reduction. Figure 7 (a) and (b) correspond to matchups with both a high similarity score and a (relatively) high predicted BABIP reduction and a low similarity score and negative BABIP reduction (the "optimized" alignment is worse than the original according to the validation technique), respectively.



(a) Robbie Ray vs Jose Altuve (b) Logan Gilbert vs Kevin Kiermeier

Figure 7: Comparing Similarity Score and BABIP Reduction

This is meant to exemplify how extreme training distributions that are a product of low similarity scores produce infeasible alignments that perform poorly in validation. While these cases show that an improvement in the SEAM methodology for matchups with low similarity may benefit the results of the fielder optimization routine, the low correlation points to another issue.

Although the similarity score cannot fully explain the variation in BABIP reduction, the lack of improvement for left-handed batters could be partially explained by sample variation and a change in approach by batters with extreme pull tendencies. If this were not the case, one would likely see a distribution similar to that of the right-handed batters for which the data is centered near a -5% predicted BABIP change with a right skew to account for said variation. It is improbable that the observed number of left-handed batters suddenly changed their approach in 2022 after showing consistent spray patterns in the 2017 to 2021 training data, pointing to a systematic issue.

Another explanation of this is how radii' shape and size variation notably influence the estimated number of balls a player can field. This brings the statistical justification of each radii implementation into question as it is difficult to gauge the true range of a fielder without sufficient data. At the time of this writing, spatial positioning data collected by the MLB is kept private.

To remedy a lack of peers with strong similarity scores, a hybrid approach may be taken for which a composite similarity score describing the aggregate similarity in the donor pool for the batter and pitcher combined, is reported. Suppose this score is low (indicating few batters and pitchers similar to those under observation). In that case, the hybrid approach may report an alignment based on the batter's marginal spray chart versus all pitchers.

Efforts to validate the optimization technique using SLUGIP were made but encountered the same biased distribution issue discussed in Section 7. Validation using SLUGIP was abandoned as basing a prediction on a prediction created from biased data would provide ambiguous results. Ideally, one would calculate the predicted slugging for each (x, y) coordinate on the field using unbiased hitting and positioning data that is solely based on how many bases the batter could advance if the ball is hit to said location. This would accurately show where each player tends to bat extra-base hits, enabling the defense to adjust their alignment accordingly.

Thus, the stochastic gradient descent maximization method tends to show a BABIP reduction in the majority of individual matchups. However, evaluating the degree of improvement is challenging due to inconsistencies in the optimization method for extreme cases, leading to impractical alignments with a greater predicted BABIP than a traditional alignment.

7 Discussion

The primary contribution of this work is to provide a statistically justified optimization model that moves players to positions on the field where BABIP is minimized. The context of the optimization enables the fielder alignment to be tailored to a particular batter-pitcher matchup, a method not shown in any other publicly recognized work. Global optimization is achieved in this context using a stochastic search method on a mathematically justified distribution with statistically justified input parameters. The stochastic search method is tuned to provide the most optimal fielder alignments while attempting to maintain a convergence rate fast enough for practical applications.

The run time of the simulated annealing method is controlled by setting bounds for the number of iterations based on the size and movement of the gradient steps, but the overall computational cost is not ideal in practice. As previously mentioned, an MLB team would have to preprocess hundreds of alignments as there would be insufficient time to calculate them during a game. A method of this pace would leave teams vulnerable to unexpected lineup changes, so having a method that could generate alignments with similar accuracy in a matter of seconds would greatly improve practicality.

In discussions on simulated annealing's computational speed, it was suggested to use a faster method like BFGS on the density beforehand to provide an approximate solution and speed up simulated annealing's convergence. The idea is that simulated annealing would not run as broad of a search since it already has an approximate solution. In practice, the simulated annealing plots could be preprocessed for starting batters while approximated alignments could be generated for batters coming off of the bench to decrease downtime for in-game optimal alignment calculations. Thus, computational costs would be saved and the time to compute a global optimal alignment for a substitute batter would be significantly decreased. One may reference Section 4.3.2 for further critiques.

In a baseball context, these optimized alignments create an improved defensive strategy that generates more outs and (subsequently) wins for the user. As noted before, this work is relevant as MLB teams spend millions of dollars each year on draft picks and free agents intending to improve their defense to manufacture wins. Each team's analytics budget is reported unofficially, but teams known to invest significantly in analytics tend to be those regularly performing well and competing in the playoffs. For example, the Tampa Bay Rays, Houston Astros, and Los Angeles Dodgers have found repeated success in the Statcast Era and have shown that deriving advantages through data and statistics improves the makeup of a high-level baseball team.

As for future directions of this work, the most practical first step is to improve computation time. This means implementing a simulated annealing method from scratch for more control over parameters and generating an approximate solution via BFGS before global optimization. As stated, this will decrease computation time and provide a solution more efficiently.

Expanding on the fielder shape rationale from Section 7, I aim to develop fielder shapes that are mathematically rigorous and better represent a player's fielding range according to their positioning as the ellipses serve as a simple approximation but do not fully reflect a fielder's real-life tendencies. It is evident in Figure 8(b) and (d) that a fielder can field the ball effectively moving towards home plate, meaning a more rigorous implementation would justify this model to a higher degree. This rigorous fielder shape may increase computation time, a factor that must be considered for practical purposes.

Incorporating slugging percentage into the distribution can be beneficial for identifying alignments that minimize this metric instead of solely BABIP. A simple way to resolve this is to take a (statistically justified) linear combination of the SEAM distribution and expected slugging percentage at each field location to create a distribution that relies on both hit frequency and slugging percentage. While straightforward, integrating SEAM with slugging percentage necessitates careful consideration to establish a mathematically sound and practical distribution.

Finally, creating an online application for this work would allow users to explore this defensive positioning model to improve public understanding of defensive shift effectiveness through statistical analysis. This work lays the foundation for many other applications regarding defensive positioning for teams, analysts, and enthusiastic fans alike with the intent to be built upon and generalized for baseball as well as other similar sports (softball, cricket, etc.) or applications (region design, network planning, ecological partitioning).

References

[1] Major League Baseball. *Baseball Savant*. Accessed on 5 June 2023. 2022. URL: https://baseballsavant.mlb.com/.

[2] David Block. *Baseball Before We Knew It*. Lincoln, NE: University of Nebraska Press, Mar. 2005.

[3] Elizabeth L. Bouzarth et al. "Swing shift: a mathematical approach to defensive positioning in baseball". In: *Journal of Quantitative Analysis in Sports* 17 (2020), pp. 47–55. URL: https://api.semanticscholar.org/CorpusID:212411628.

[4] R. Church and C.S. Revelle. "The maximal covering location problem". In: *Papers of the Regional Science Association* 32 (1974), pp. 101–118.

[5] Cameron Grove. *The Shift Tester*. Accessed on 24 January 2023. 2022. URL: https://pitching.shinyapps.io/ShiftTester/.

[6] Trevor Hastie and Robert Tibshirani. *Generalized additive models*. Wiley Online Library, 1990.

[7] A.T. Murray, A. Ortiz, and S. Cho. "Enhanced strategic defensive positioning and performance in the outfield". In: *Journal of Geographical System* 34 (Jan. 2022), pp. 223–240.

[8] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd ed. Springer, 2006.

[9] R Core Team. *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria, 2013. URL: http://www.R-project.org/.

[10] R.W. Swain and C.S. ReVelle. "Central facilities location". In: *Geographical Analysis* 2 (1970), pp. 30–42.

[11] Julia Wapner, David Dalpiaz, and Daniel J. Eck. *SEAM methodology for context-rich player matchup evaluations*. 2022. arXiv: 2005.07742 [stat.AP].

[12] Marvin N. Wright and Andreas Ziegler. "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R". In: *Journal of Statistical Software* 77.1 (2017), pp. 1–17. DOI: 10.18637/jss.v077.i01.

[13] Yang Xiang et al. "Generalized Simulated Annealing for Efficient GlobalOptimization: the GenSA Package for R." In: *The R Journal* 5/1 (2013). URL: https://journal.r-project.org.

Appendix A

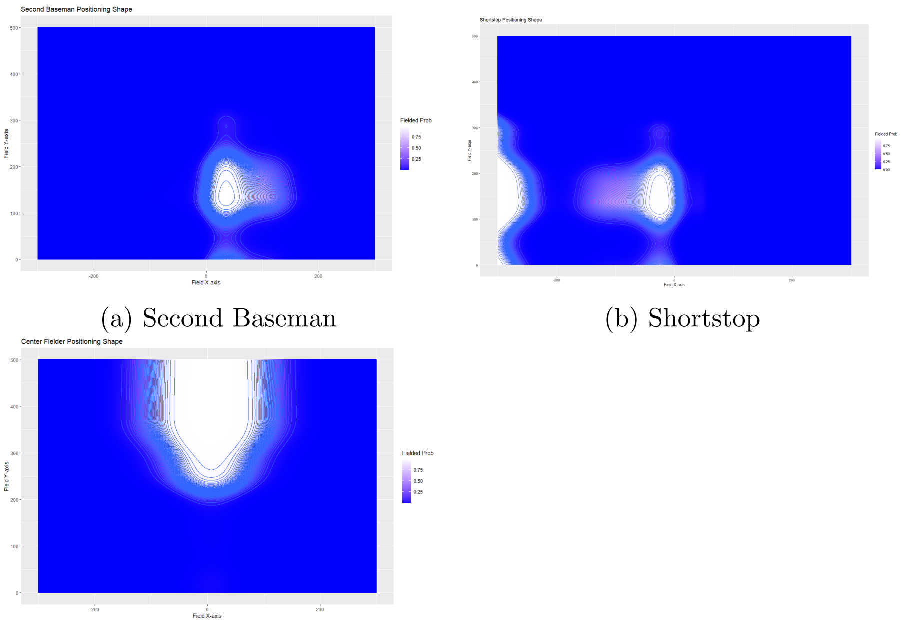
A.1 Fielder Positioning Shapes

To determine whether using elliptical fielder ranges is a reasonable assumption, Generalized Additive Models (GAM or GAMs) (Hastie and Tibshirani [6]) were fit to balls in play via Baseball Savant data to determine the fielder range shape at each position.

GAM training was restricted to balls in fair territory as batters cannot advance the bases by hitting a ball in foul territory, making foul balls irrelevant to minimizing BABIP or SLUGIP. Two methods were used to generate these shapes: 1. A ball in play was marked with a “1” at the coordinate for which the fielder touched the ball and “0” otherwise. 2. For balls in play resulting in an out, a “1” was marked at the coordinate for which the fielder fielded the ball and “0” otherwise. The resulting plots represent the percent of balls fielded by a player at each coordinate restricted to the fielder position (first base, short, etc.) and the percent of balls fielded by a player at each coordinate resulting in an out restricted to the fielder position, respectively.

The methods yielded nearly identical shapes; the latter is shown in Figure 8. By observation, the middle infielders have a cone-like shape with a roughly elliptic “tail” on the opposite side of second base; the center fielder has a symmetric cone shape facing home plate. A relatively similar phenomenon was expected for corner fielders, but the vast number of foul balls fielded in foul territory skewed these range shapes. This results in a fielder range that is not a standard geometric shape, making it more difficult to justify.

Although the corner fielders have a non-standard geometric range, they were assumed to be elliptical for implementation simplicity as balls hit down the foul line are difficult to defend in any case. In baseball, it is uncommon to position a fielder near a foul line as this would create gaps of vulnerability in the defensive coverage between fielders.



(a) Second Baseman

(b) Shortstop

(c) Center Fielder

Figure 8: Plots of Fielder Positioning Shapes

A.2 Conditional Distribution

To create a distribution for which the optimization depends on slugging percentage, the aim was to combine the SEAM distribution with a distribution of predicted slugging percentage based on ball-flight metrics. This would allow tailoring of the algorithm to minimize slugging percentage.

The predicted distribution of slugging was trained on balls in play from Baseball Savant [1] data.

First, launch angle (LA) and exit velocity (EV) were predicted at each coordinate for a given batter. Using the predicted LA and EV values, the probability of each batted ball event (out, single, double, etc.) was calculated at each coordinate to find the location's predicted slugging. As a toy example, a ball hit near the outfield fence may have a probability distribution as follows:

$$\begin{cases} \mathbb{P}(\text{out}) = 0.40 \\ \mathbb{P}(\text{single}) = 0.10 \\ \mathbb{P}(\text{double}) = 0.20 \\ \mathbb{P}(\text{triple}) = 0.10 \\ \mathbb{P}(\text{home run}) = 0.20 \end{cases}$$

Then the expected slugging percentage for the coordinate would be the following:

$$\begin{aligned} \mathbb{E}(\text{slugging}) &= (0 \text{ bases}) * \mathbb{P}(\text{out}) + (1 \text{ base}) * \mathbb{P}(\text{single}) + (2 \text{ bases}) \\ &\quad * \mathbb{P}(\text{double}) + (3 \text{ bases}) * \mathbb{P}(\text{triple}) + (4 \text{ bases}) * \mathbb{P}(\text{home run}) \\ &= 0 * 0.4 + 1 * 0.1 + 2 * 0.2 + 3 * 0.1 + 4 * 0.2 = 1.4 \end{aligned}$$

These predictions were generated using the random forest **ranger** (Wright and Ziegler [12]) package in R.

In addition, an attempt to incorporate the base-running ability of a batter was made by calculating his on-base percentage (OBP) and using it as a predictor in the random forest model; OBP is insignificant in this prediction. The rationale for incorporating OBP is that a fielder's positioning may be affected by the batter's baserunning ability. For instance, middle infielders may need to play closer to home plate for faster batters, enabling them to field the ball and make a quicker throw to first base.

After calculating the distribution, the predicted slugging value was multiplied by the SEAM predicted batted ball value at each coordinate to produce a distribution reflecting both SEAM and predicted slugging percentage.

Counterintuitively, areas of the field hypothesized to exhibit greater "z" values are relatively low. To illustrate, the outfield was hypothesized to yield greater combined values since balls hit to the outfield (that are not fielded) tend to result in extra-base hits. The opposite held true as outfielders can field most flyballs in their vicinity, resulting in the majority of outfield locations having notably low "z" values.

The resulting distribution is counterintuitive for our optimization because the expected slugging distribution is already conditioned on where fielders are typically positioned since a fielder is more likely to create an out on a ball hit near them. Thus, this combined distribution is unusable as the fielders would be placed in areas of the field for which fewer balls hit, resulting in a higher expected slugging percentage for any given batter.

Appendix B

The following table shows the percent change in BABIP for all tested players.

Table 4: BABIP Change via Repositioning

	Pitcher	**Batter**	**BABIP % Change**
1	Logan Gilbert	Jose Altuve	-3.4563
2	Merrill Kelly	Jose Altuve	-4.9811
3	Kyle Wright	Jose Altuve	-6.1719
4	Jordan Montgomery	Jose Altuve	-3.2530
5	Framber Valdez	Jose Altuve	-5.6927
6	Robbie Ray	Jose Altuve	-5.6637
7	Logan Gilbert	Nolan Arenado	-2.6430
8	Merrill Kelly	Nolan Arenado	-5.1409
9	Kyle Wright	Nolan Arenado	-5.7653
10	Jordan Montgomery	Nolan Arenado	-0.6535
11	Framber Valdez	Nolan Arenado	-2.1493
12	Robbie Ray	Nolan Arenado	-4.4583
13	Logan Gilbert	Lorenzo Cain	-5.0828
14	Merrill Kelly	Lorenzo Cain	-5.5039
15	Kyle Wright	Lorenzo Cain	-6.7238
16	Jordan Montgomery	Lorenzo Cain	-3.9936
17	Framber Valdez	Lorenzo Cain	-6.6947
18	Robbie Ray	Lorenzo Cain	-5.1990
19	Logan Gilbert	Evan Longoria	0.8423
20	Merrill Kelly	Evan Longoria	-4.2695
21	Kyle Wright	Evan Longoria	-5.2570
22	Jordan Montgomery	Evan Longoria	-2.6576
23	Framber Valdez	Evan Longoria	-4.2405
24	Robbie Ray	Evan Longoria	-6.2881
25	Logan Gilbert	Albert Pujols	-4.9376
26	Merrill Kelly	Albert Pujols	-6.5205
27	Kyle Wright	Albert Pujols	-2.6430
28	Jordan Montgomery	Albert Pujols	-4.7488
29	Framber Valdez	Albert Pujols	-4.5455
30	Robbie Ray	Albert Pujols	-2.9335

Continued on next page

Table 4 (cont.)

	Pitcher	Batter	BABIP % Change
31	Logan Gilbert	Mike Trout	-3.7613
32	Merrill Kelly	Mike Trout	-5.4458
33	Kyle Wright	Mike Trout	-6.2736
34	Jordan Montgomery	Mike Trout	-3.8629
35	Framber Valdez	Mike Trout	1.6410
36	Robbie Ray	Mike Trout	-5.7944
37	Logan Gilbert	Hunter Renfroe	-1.1908
38	Merrill Kelly	Hunter Renfroe	-3.9065
39	Kyle Wright	Hunter Renfroe	-3.3692
40	Jordan Montgomery	Hunter Renfroe	-2.1202
41	Framber Valdez	Hunter Renfroe	-3.0351
42	Robbie Ray	Hunter Renfroe	-5.9396
43	Logan Gilbert	Trey Mancini	-3.1658
44	Merrill Kelly	Trey Mancini	-5.4023
45	Kyle Wright	Trey Mancini	-5.5184
46	Jordan Montgomery	Trey Mancini	-4.9811
47	Framber Valdez	Trey Mancini	-4.6326
48	Robbie Ray	Trey Mancini	-5.3587
49	Logan Gilbert	Vladimir Guerrero	1.1037
50	Merrill Kelly	Vladimir Guerrero	-5.2861
51	Kyle Wright	Vladimir Guerrero	-4.5745
52	Jordan Montgomery	Vladimir Guerrero	0.8132
53	Framber Valdez	Vladimir Guerrero	-0.5954
54	Robbie Ray	Vladimir Guerrero	-5.4604
55	Logan Gilbert	Taylor Ward	2.8754
56	Merrill Kelly	Taylor Ward	-5.6927
57	Kyle Wright	Taylor Ward	-1.0601
58	Jordan Montgomery	Taylor Ward	-3.8484
59	Framber Valdez	Taylor Ward	-4.4148
60	Robbie Ray	Taylor Ward	-5.5475
61	Logan Gilbert	Matt Carpenter	2.8754
62	Merrill Kelly	Matt Carpenter	-0.7987
63	Kyle Wright	Matt Carpenter	0.3485
64	Jordan Montgomery	Matt Carpenter	-2.1348

Continued on next page

Table 4 (cont.)

	Pitcher	Batter	BABIP % Change
65	Framber Valdez	Matt Carpenter	2.9916
66	Robbie Ray	Matt Carpenter	-1.6120
67	Logan Gilbert	Freddie Freeman	-0.9439
68	Merrill Kelly	Freddie Freeman	2.3526
69	Kyle Wright	Freddie Freeman	0.3776
70	Jordan Montgomery	Freddie Freeman	4.1824
71	Framber Valdez	Freddie Freeman	0.8713
72	Robbie Ray	Freddie Freeman	0.0436
73	Logan Gilbert	Joey Gallo	-1.3651
74	Merrill Kelly	Joey Gallo	-0.3485
75	Kyle Wright	Joey Gallo	1.8008
76	Jordan Montgomery	Joey Gallo	1.3070
77	Framber Valdez	Joey Gallo	3.7032
78	Robbie Ray	Joey Gallo	-3.1513
79	Logan Gilbert	Dee Strange-Gordon	-6.9561
80	Merrill Kelly	Dee Strange-Gordon	-6.0558
81	Kyle Wright	Dee Strange-Gordon	-7.0578
82	Jordan Montgomery	Dee Strange-Gordon	-2.0767
83	Framber Valdez	Dee Strange-Gordon	-5.5039
84	Robbie Ray	Dee Strange-Gordon	-4.4293
85	Logan Gilbert	Bryce Harper	-2.4688
86	Merrill Kelly	Bryce Harper	-3.7903
87	Kyle Wright	Bryce Harper	1.9895
88	Jordan Montgomery	Bryce Harper	6.1429
89	Framber Valdez	Bryce Harper	3.4127
90	Robbie Ray	Bryce Harper	0.0436
91	Logan Gilbert	Jason Heyward	-2.1929
92	Merrill Kelly	Jason Heyward	-4.3421
93	Kyle Wright	Jason Heyward	1.6265
94	Jordan Montgomery	Jason Heyward	1.3506
95	Framber Valdez	Jason Heyward	-2.6721
96	Robbie Ray	Jason Heyward	0.6680
97	Logan Gilbert	Kevin Kiermaier	3.2965
98	Merrill Kelly	Kevin Kiermaier	0.7842

Continued on next page

Table 4 (cont.)

	Pitcher	Batter	BABIP % Change
99	Kyle Wright	Kevin Kiermaier	-0.0726
100	Jordan Montgomery	Kevin Kiermaier	0.8132
101	Framber Valdez	Kevin Kiermaier	-1.3506
102	Robbie Ray	Kevin Kiermaier	0.6680
103	Logan Gilbert	Anthony Rizzo	-1.7572
104	Merrill Kelly	Anthony Rizzo	-0.7697
105	Kyle Wright	Anthony Rizzo	-2.1057
106	Jordan Montgomery	Anthony Rizzo	3.1658
107	Framber Valdez	Anthony Rizzo	3.8920
108	Robbie Ray	Anthony Rizzo	2.3962
109	Logan Gilbert	Joey Votto	-0.3485
110	Merrill Kelly	Joey Votto	-0.2178
111	Kyle Wright	Joey Votto	-3.1078
112	Jordan Montgomery	Joey Votto	2.9335
113	Framber Valdez	Joey Votto	2.0331
114	Robbie Ray	Joey Votto	1.8734
115	Logan Gilbert	Tony Kemp	-0.6245
116	Merrill Kelly	Tony Kemp	-0.8568
117	Kyle Wright	Tony Kemp	-0.3050
118	Jordan Montgomery	Tony Kemp	-0.4357
119	Framber Valdez	Tony Kemp	1.3070
120	Robbie Ray	Tony Kemp	-0.3776