Journal of Quantitative Analysis in Sports

Volume 5, *Issue* 2 2009 *Article* 8

2008 NORTHERN CALIFORNIA SYMPOSIUM ON STATISTICS AND OPERATIONS RESEARCH IN SPORTS

Using Simulation to Estimate the Impact of Baserunning Ability in Baseball

Ben S. Baumer*

*CUNY Graduate School and University Center, bbaumer@gc.cuny.edu

Using Simulation to Estimate the Impact of Baserunning Ability in Baseball*

Ben S. Baumer

Abstract

In baseball, an offensive team's run scoring ability is dependent not only upon the batting skills of its players, but also their baserunning abilities. Using a Monte Carlo simulation based on actual statistics of real players, we estimate the magnitude of the effect of baserunning skills upon a team's run scoring ability. Our results largely confirm previous non-academic estimates that the impact of baserunning upon a team's run scoring ability is typically less than ± 25 runs per season. However, we show using simple heuristic algorithms, that a team composed of the best (worst) nine baserunners could gain (lose) as many as 70 (55), runs per season due to baserunning.

KEYWORDS: baseball, baserunning, simulation, Monte Carlo, sports

*The author was supported by the Susan Rosenberg Zalk Student Travel and Research Fund Award. The author is indebted to the the New York Mets for the use of their databases and processor cycles.

1 Introduction

1.1 Motivation

Recently, estimating the run scoring abilities of Major League baseball teams has become something of a sport unto itself, being central to the strategy of the sabermetrically oriented front office of the Oakland A's as depicted in *Moneyball* [8]. To this end, the offensive contributions of position players have been greatly illuminated through the use of statistical analysis and discrete event simulation. But while the batting abilities of players have been studied extensively, measuring the effect of skillful baserunning has proven more elusive. Clearly, the process of scoring runs depends upon both the batting and baserunning abilities of offensive players, with the contributions of the former certainly dwarfing the latter. The purpose of this paper is to bound the potential impact of the latter.

Let us be clear at this point that we consider baserunning to incorporate everything that happens to the player apart from the outcome of his plate appearance. In particular, we are just as interested in the frequency with which a batter "takes the extra base" (i.e. - advances more than one base on a single) as the frequency with which he steals a base. We also consider relevant the frequency with which he is doubled up on a groundout with a runner on first and less than two outs, and the frequency with which he tags up and scores on a flyball out with less than two outs.

1.2 Literature Review

Our approach to the problem was to create an accurate model of offensive player movement using discrete event simulation, and then experiment with different parameters to measure the impact of baserunning. Existing research relevant to this subject[1], broke down into two camps.

In the first, there were two prominent non-academic articles published in 2005 that attempted to calculate the number of runs gained or lost by individual baserunners using actual play-by-play data. In both cases, the approach was based on summing the value of changes to the expected run matrix[2] caused by the movement of actual baserunners in actual games. Specifically, Click [4] calculated the number of extra bases gained by individual baserunners over a 32 year period, finally concluding that an individual baserunner could increase his team's run total by as much

[1] Turocy [13] studied the value of base-stealing from an economic perspective.

[2] It is common in such articles to refer to the "Expected Run Matrix". This is a matrix $M \in \mathbb{R}^{8,3}$, where each entry m_{ij} represents the expected number of runs that will score in the remainder of the inning, given that there are currently j outs, and the configuration of the baserunners is coded by i. Here i represents any of the $8 = 2^3$ possible baserunner configurations.

as 10 runs over the course of a full season. In a series of articles published online shortly thereafter, Fox [6] used a similar framework to estimate that the number of runs gained or lost due to baserunning for a team over the course of a season was around 20.

In the second camp, we found simulation-based academic research, which had heretofore focused primarily on batting order optimization using Markov chain models. The first to use this method appears to have been Freeze [7], who showed that batting order was relatively unimportant, with the difference between the best and worst orders amounting to less than three extra wins over a 162-game season. But Freeze was hardly the last to use this approach, as Bukiet, Harold, and Palacios [3] constructed a flexible and extendable Markov model, which was used by Sokol [10] to determine heuristics for batting order optimatization. In fact, the Markov model is so naturally applicable to baseball that Sokol claims it was independently proposed by not only Bukiet, Harold, and Palacios, but also Pankin [9], Thorn and Palmer [12], Cook [5], and others. Albert [1] provides a simpler, more accessible version of this model.

We decided that a Markov model was unsuitable for our purposes, as each transition matrix would now depend not only on the batter, but also the baserunners. While this would, in fact, be ideal if sufficient data existed, we found that the sample sizes of individual baserunner-batter combinations were much too small to make meaningful statistical inferences.

Yet it is clear that in reality, the abilities of both the batter *and* the baserunners affect the advancement of the baserunners on a particular ball in play. To see this, consider the likelihood of a runner scoring from second base on a single. If the batter is the slap-hitting Luis Castillo, such a single might very well be of the infield variety, in which case the runner may not advance at all. On the other hand, if the batter is the slugging David Ortiz, it could be a line drive off of the outfield wall, in which case the runner may easily score. Conversely, consider how that likelihood is affected by the identity of the baserunner, who might be the speedy Jose Reyes, or the plodding Frank Thomas. To accurately capture the former phenomenon, we would need to simulate the location and trajectories of batted balls, which again is difficult to do without slicing the data down to unusably small samples. In this paper, we use simulation to model the latter phenomenon.

So, of the two camps of existing research that we indentified, the first is sound in its attempt to estimate the impact of baserunning ability upon a team's run scoring performance, but lacks the flexibility to carry out experiments that we gain from simulation. As for the second camp, Markov chains are unfeasible for this application given the demands they would place on the data. Thus, our approach is to build a discrete event simulation based on the statistics of actual players, in order to quantify the effect of baserunning ability upon team run scoring.

DOI: 10.2202/1559-0410.1174

1.3 Structure of this Paper

In Section 2, we give a full explanation of our model, including both a mathematical description, as well as some details about its technical implementation. In Section 3, we give our results, which come in three parts. First, we test the accuracy of our model by comparing the distribution of runs scored per half-inning culled from actual MLB statistics, with those generated by our model with a league average lineup programmed in. This validates the soundness of our testing environment. Second, we examine the changes in the distribution of runs scored per team when the actual baserunning statistics for individual players are input, versus when league average baserunning values are used for all players. This provides us with an average case estimate for the impact of baserunning upon run scoring. Third, using some simple heuristic algorithms, we attempt to construct lineups which maximize (minimize) the impact of individual baserunning, and examine the magnitude of the difference upon run scoring. This establishes upper (lower) bounds for the change in run scoring ability that a team can hope to achieve through baserunning. In Section 4, we summarize our findings and offer suggestions for future research.

2 The Model

2.1 The State of the System

Let $\mathscr{P}$ be the set of all major league players. From this set, we choose a lineup $L \in \binom{\mathscr{P}}{9}$ of nine players, and specify a batting order $\pi \in \Pi(\mathbb{Z}_9)$, which is simply a permutation on those players. This ordered lineup forms an offensive team, $T = T(L, \pi)$. Each team T takes part in a half-inning, and what interests us is the number of runs, R_T, that they score before recording three outs. Thus, we simulate a half-inning by beginning with a team T and an integer $k \in \mathbb{Z}_9$, indicating the lineup position currently at-bat, and running the simulation until three outs are recorded. At each stage of the simulation the k^{th} player in $\pi(L)$ comes to bat, a batting event is sampled, and the baserunners move probabilistically according to rules that are described below. In addition to R_T, we must keep track of $O \in \mathbb{Z}_3$, the number of outs in the half-inning, and $l \in \mathbb{Z}_4^9$, a vector that records the location of each of the nine batters in the lineup (i.e. - on any of the three bases, or in the dugout)[3]. At the end of the half-inning, R_T is output.

[3] Note that the number of batters who have come to the plate in a half-inning is always equal to the number of runners on base plus the number of runners who have scored, plus the number of outs. This serves as a useful check.

2.2 The Player Specification

Let Ω be the set of all events that can occur at any given moment in a baseball game. A realistic description of this set would exceed the limits of this paper, so in order to make our simulation feasible, we restrict our attention to a subset $M \subset \Omega$. Specifically, let

$$M = \{1B, 2B, 3B, HR, BB, HBP, SO, GO, AO, WP, SB, CS, E\}$$

be the events we allow in our simulation, where GO is a ground out, AO is an air out, and all other abbreviations are consistent with standard baseball terminology. Moreover, we define $A = \{1B, 2B, 3B, HR, BB, HBP, SO, GO, AO\} \subset M$ to be a mutually exclusive set of outcomes of a plate appearance. Now, for each batter $P \in L$, we load statistics from a database about the relative frequency of events in A for that player. These statistics define an empirical multinomial distribution over A, with probability density function f_P. For the purposes of our simulation, f_P encapsulates everything we know about player P's batting ability.

2.2.1 Baserunner Advancement

Similarly, with the identities and proclivities of the baserunners known, advancement also occurs via Monte Carlo processes. In particular, let $B \subseteq \Omega \setminus A$ be a set of non-batting related events. Specifically, we include in B eight different ways to advance on the bases:

1. Advance from 1st to 3rd on a single
2. Score from 1st on a double
3. Score from 2nd on a single
4. Beat out a double play attempt on a ground out with a runner on 1st and less than 2 outs
5. Steal 2nd when that base is unoccupied
6. Steal 3rd when that base is unoccupied
7. Tag up from either 2nd or 3rd on a fly out with less than 2 outs in the inning

Each can be thought of as a different way to ”take the extra base.” It is clear from both conventional wisdom and statistical analysis that individual baserunners can have wildly different rates of both attempting and succeeding to advance in these ways.

So, for each player P, we define two vectors $Y_P, Z_P \in [0,1]^8$, where each entry in Y_P consists of the probability of attempting to advance in one of the eight ways listed above, and Z_P consists of the conditional probability of successfully advancing in each of the eight cases, given an attempt. For example, the MLB averages for the 2005-2007 seasons were[4]:

$$\hat{Y} = \begin{pmatrix} Adv13 \\ Adv14 \\ Adv24 \\ GIDP \\ Steal2 \\ Steal3 \\ TagUp2 \\ TagUp3 \end{pmatrix} = \begin{pmatrix} 0.2554 \\ 0.4105 \\ 0.5768 \\ 0.6299 \\ 0.0585 \\ 0.0130 \\ 0.2371 \\ 0.5540 \end{pmatrix}, \qquad \hat{Z} = \begin{pmatrix} 0.9576 \\ 0.9282 \\ 0.9466 \\ 0.3556 \\ 0.6995 \\ 0.6855 \\ 0.8721 \\ 0.9571 \end{pmatrix}$$

The entry $\hat{Y}_{(1)} = 0.2554$ indicates that during this time period, after the batter hits a single, the runner on first base attempted to advance to third base about 26% of the time, while the entry $\hat{Z}_{(1)} = 0.9576$ indicates that such runners were successful in about 96% of those instances.

The description of grounding into a double play warrants further explanation. In this framework, any ground out with a runner on first base and less than two outs is viewed as a potential double play, just as any plate appearance with a runner on first and no one on second base is viewed as a potential steal of second. The entry $\hat{Y}_{(4)} = 0.6299$ indicates that about 63% of the time when a ground out is hit with a runner on first and less than two outs, the runner on first is forced at second base. Next, the entry $\hat{Z}_{(4)} = 0.3556$ indicates that in about 36% of *those* instances, the batter made it safely to first (i.e. - he advanced). Thus, of all ground outs with a runner on first and less than two outs, about 37% result in the runner advancing and the batter being forced at first, 22% result in a fielder's choice at second, and 41% result in a double play.

Thus, in our simulation, every player P is defined by a triple (f_P, Y_P, Z_P) consisting of a probability density function f_P over A, which describes the player's batting ability, and two vectors in the unit hypercube, the first of which, Y_P, determines the player's baserunning aggressiveness, while the second, Z_P, determines his baserunning success.

2.3 Constants and Assumptions

Further baserunner advancment is applied uniformly in the simulation, without regard to the identity of the baserunners. Namely, we define three empirically-

[4] All data for this study was furnished by STATS, Inc.

determined constants:

1. BIP-ERROR-RATE = 0.0140, the rate of fielding errors on balls put into play.
2. SB-ERROR-RATE = 0.0706, the rate of fielding errors on stolen base and pick-off attempts.
3. WILD-PITCH-RATE = 0.0230, is the rate of wild pitches, passed balls and balks per plate appearance. They are so grouped since they all have the same consequences, and are similarly assumed to be random events that are independent of the players in question.

Each of these events result in all baserunners advancing one base.

Additional simplifying assumptions to possible baserunner advancements were made. For example, when the batter grounds into a double play or a fielder's choice, the force play is always at second base (i.e. - a runner cannot be forced out at third base or home plate). In non-double play situations, the baserunners always advance on a ground out. If the runner on second attempts to steal third, then if there is a runner on first, he automatically advances to second (i.e. - a double steal is implied). A runner cannot tag up from first and go to second. Otherwise, runners advance only when forced[5]. If a player singles in a run, he never advances to second on the throw home. Fielding positions are ignored, and the designated hitter rule is in effect for all teams.

2.4 The Greedy Batting Order Algorithm

By default, for each major league team, we populated the lineup with the nine players on the team's active roster on August 28th, 2008 with the most cumulative plate appearances during the 2005-2007 seasons. Given this lineup $L \in \binom{\mathscr{P}}{9}$ of nine batters, we used a simple greedy algorithm to set a default batting order $\pi_0(L)$. The algorithm starts by placing the player with the most cumulative plate appearances overall in the lineup spot in which he has had the most plate appearances. Then, we move to the player with the second most total plate appearances, and put him in the spot among those remaining in which he has had the most plate appearances. If a player does not have any plate appearances in any of the available unoccupied lineup spots, one is randomly chosen for him. This process continues until the lineup is set. In effect, we create a matching between the set of the nine players and the set of the nine spots in the batting order.

The purpose of this greedy algorithm is simply to fix a reasonable default batting order for any nine players, and no claims about its accuracy or effectiveness will be made.

[5] That is, according to ”ghostrunner” rules.

DOI: 10.2202/1559-0410.1174

2.5 The Output

To isolate the effects of baserunning, the simulation is capable of running in two modes:

- `BASIC-MODE`: Every player is assigned baserunning probabilities according to the major league averages. That is $Y_P = \hat{Y}$ and $Z_P = \hat{Z}$ for all $P \in \mathscr{P}$.
- `REAL-MODE`: Every player is assigned baserunning probabilities based on his actual baserunning statistics. That is, for every $P \in \mathscr{P}$, Y_P and Z_P are determined empirically, in the manner described above.

In both cases, every player is assigned batting probabilities f_P based on his actual batting statistics.

For any team T, let $\overline{R_T}$ be the average number of runs scored per half-inning by team T in a simulation run in `BASIC-MODE`. Similarly, let $\overline{R_T^*}$ be the average number of runs scored by team T in `REAL-MODE`. Then the average difference due to baserunning per half-inning is $D_T = \overline{R_T^*} - \overline{R_T}$. The quantity $X_T = D_T \cdot 9 \cdot 162$ gives us an estimate of the number of additional runs scored due to individual baserunning ability per team, per season[6].

For each team T, we will be interested in the distribution of runs scored per half-inning under various scenarios, but in general, our primary focus will be to estimate X_T, the difference in run scoring ability due to baserunning over the course of a season.

3 The Results

3.1 Testing the Accuracy of the Model

In all calculations, we restrict our data set to regular season Major League baseball games played between 2005 and 2007. The average number of runs scored per half-inning in the Major Leagues over this time period was 0.531. Meanwhile, the percentage of half-innings in which at least one run was scored was about 28.7%. To validate our simulation, we loaded the MLB average batting statistics for each of the nine lineup spots, and ran the simulation in `BASIC-MODE`[7]. By running our simulation on this team ($\hat{T}$), we found that the distribution of runs scored returned by our simulation was fairly accurate. The results of our simulation are shown in Table 1.

[6] Since there are 9 innings in a game and 162 games in a season.

[7] The simulations are fast! We can simulate 1 million half-innings in under 27 seconds.

Source	numInnings	$\bar{R}$	SE	$s^2(R)$	$Pr[R>0]$	$Pr[R>1]$
MLB Actual	131,342	0.531		1.125	0.287	0.134
Simulation 1	1,000,000	0.528	0.001	1.095	0.286	0.136
Simulation 2	1,000,000	0.529	0.001	1.100	0.286	0.136
Simulation 3	1,000,000	0.531	0.001	1.104	0.287	0.136

Table 1: Comparison of Runs Scored, Actual vs. Simulated

We can see that most of the time, a 95% confidence interval for $\overline{R_{\hat{T}}}$ contained the true mean (0.531). However, it is true that the variance of $R_{\hat{T}}$ in our model was consistently lower than that of major league baseball.

3.2 Testing the Cumulative Effect of Baserunning on a Team

Next, we ran the simulation one million times for each team in MLB, and recorded X_T. Table 2 reveals that the maximum value of X that we found was 21.1, for the New York Mets, while the minimum value was -20.8, for the Chicago White Sox. The complete results, along with p-values for a two-tailed, two-sample t-test, are shown in Table 2.

In all, eight teams demonstrated positive baserunning ability at the 5% significance level, while eleven teams demonstrated statistically significant negative baserunning ability. These results are very much in line with the estimates of Click and Fox, while also passing a ”sniff test.”[8]

3.3 Maximizing the Impact of Baserunning on a Team

We have seen that for a typical team, $|X| \leq 25$. But can we find a greatest lower bound for $\max|X|$? Our goal in this section is to find a team that maximizes $|X|$.

Recall that $\mathscr{P}$ represents the set of all major league players. With 25 players on each of 30 teams, we have that at any given time, $|\mathscr{P}| = 750$. Since there are $\binom{750}{9}$ different ways to choose nine players, and an additional 9! ways to order the players in any of these choices, there are a total of

$$\binom{\mathscr{P}}{9} \cdot 9! = \frac{750!}{741!} = (750)_{(9)} = 750 \cdot 749 \cdots 742 \approx 7.155 \times 10^{25}$$

[8]The Mets and Rockies are known to have several excellent baserunners (Reyes, Beltran, Taveras), as we will see below, while the White Sox and Red Sox are reputed to have more plodding offenses.

DOI: 10.2202/1559-0410.1174

T	R_T	$s^2(R_T)$	R_T^*	$s^2(R_T^*)$	D_T	X_T	p-value
Mets	0.594	1.269	0.609	1.277	0.014	21.1	0
Yankees	0.686	1.523	0.698	1.547	0.012	18.0	0
Rockies	0.545	1.148	0.556	1.153	0.010	14.7	0
Reds	0.370	0.694	0.379	0.707	0.008	12.1	0
Mariners	0.488	0.988	0.496	0.996	0.008	11.6	0
Nationals	0.475	0.961	0.479	0.974	0.005	7.1	0.001
Rays	0.585	1.264	0.589	1.274	0.005	6.7	0.004
Royals	0.489	0.988	0.493	0.992	0.004	6.1	0.003
Twins	0.471	0.958	0.473	0.962	0.003	3.7	0.065
Brewers	0.564	1.199	0.567	1.197	0.003	3.7	0.104
Pirates	0.526	1.106	0.528	1.113	0.001	1.9	0.376
`MLB-Average`	0.528	1.095	0.529	1.100	0.001	1.7	0.422
Athletics	0.499	1.049	0.500	1.049	0.001	1.0	0.652
Giants	0.449	0.885	0.449	0.875	0	0.6	0.772
Angels	0.626	1.332	0.626	1.318	0	0	0.999
Diamondbacks	0.569	1.214	0.567	1.209	-0.001	-1.8	0.422
Dodgers	0.585	1.237	0.583	1.230	-0.002	-2.6	0.259
Indians	0.611	1.325	0.609	1.323	-0.002	-3.1	0.186
Phillies	0.650	1.414	0.648	1.410	-0.002	-3.6	0.147
Tigers	0.660	1.445	0.657	1.445	-0.003	-4.1	0.097
Braves	0.599	1.280	0.596	1.280	-0.003	-4.6	0.048
Astros	0.516	1.075	0.513	1.078	-0.003	-5.1	0.018
Padres	0.410	0.805	0.405	0.796	-0.005	-6.6	0
Cardinals	0.538	1.133	0.533	1.115	-0.005	-7.5	0.001
Orioles	0.572	1.212	0.567	1.200	-0.006	-8.4	0
Cubs	0.610	1.294	0.604	1.274	-0.007	-9.6	0
Blue Jays	0.558	1.179	0.551	1.160	-0.008	-11.0	0
Rangers	0.547	1.140	0.539	1.125	-0.008	-12.2	0
Marlins	0.580	1.224	0.572	1.195	-0.008	-12.3	0
Red Sox	0.647	1.422	0.639	1.418	-0.009	-12.7	0
White Sox	0.616	1.300	0.602	1.283	-0.014	-20.8	0

Table 2: X_T for MLB teams, with p-value for a Two-Tailed, Two-Sample t-test, $n = 1000000$

possible ”teams” that could be formed (on any given day). Let $\mathscr{T} = \binom{\mathscr{P}}{9} \times \Pi_9$ be this set. We seek to maximize X_T for some team $T \in \mathscr{T}$. That is, we are interested in the theoretical quantity X^*, where

$$X^* = \max_{T \in \mathscr{T}} \left\{ X_T : T \in \binom{\mathscr{P}}{9} \times \Pi_9 \right\}$$

and similarly X_*, the minimum value of X. Clearly, the set $\mathscr{T}$ is too large to make a brute force computation efficient, and no optimization algorithm seems obvious. Accordingly, we focus on heuristics that approximate X^* by finding those players who baserunning abilities are ”best,” putting them in a lineup, and calculating X_T for this team. In the next section we detail our approach to finding such heuristics.

3.3.1 Heuristic Algorithms

Let G be a set of real-valued functions on $\mathscr{P}$. We seek a function $g \in G$ that identifies nine players $P \in \mathscr{P}$ such that a team comprised of these players maximizes $|X|$, for some batting order. However, at this point, we are much less concerned with the effects of lineup optimization (i.e. - optimizing with respect to the permutations in Π_9), and accordingly limit ourselves to finding a good set of players[9]. Specifically, for

$$X^*|_{\pi_0} = \max_{L \in \binom{\mathscr{P}}{9}} \{X_T : T = (L, \pi_0)\}$$

let $g^* : \mathscr{P} \to \mathbb{R}$ be the function that picks L such that L maximizes X over all lineups using the default batting order π_0 chosen by the greedy algorithm. To this end, we give several simple heuristics and the results of running them on our data set.

First, consider $1 - Y_{P(4)}$, the probability of being forced at first base after grounding out with a runner on first base and less than two outs. We want to use this as a measure of ”aggressiveness,” instead of $Y_{P(4)}$, since a higher value of $Y_{P(4)}$ suggests a worse outcome for the batter-runner. So let e_4 be the 4th standard basis vector, and set $Y_P{}^* = Y_P + (1 - Y_{P(4)})e_4$. That is, $Y_P{}^*$ is Y_P with $1 - Y_{P(4)}$ in place of $Y_{P(4)}$. This enables us to compute norms on $Y_P{}^*$ that measure baserunner aggressiveness, since now for each entry, a higher value indicates a greater probability of the runner attempting to advance.

Similarly, recall that Z_P measures a baserunner’s success when he attempts to advance. Thus, in each entry in Z_P, a higher value indicates a higher probability of reaching safely on an attempt to advance.

[9]Note that Freeze’s work suggests that $|X| \leq 30$ for lineup optimiziation.

DOI: 10.2202/1559-0410.1174

3.3.2 Some Simple Heuristics

The following heuristics are natural within our framework:

1. Naive Aggressive Heuristic: We define baserunner aggressiveness by simply taking the usual Euclidean norm of $Y_P{}^*$. Thus, $g_1(P) = ||Y_P{}^*||_2$.

2. Naive Conservative Heuristic: We define the most conservative baserunners as those who are thrown out least frequently. Thus, we simply take the usual Euclidean norm of Z_P. Thus, $g_2(P) = ||Z_P||_2$.

3. Naive Mixed Heuristic: A combination of the above, but we take the inner produce of $Y_P{}^*$ and Z_P. Thus, $g_3(P) = (Y_P{}^*)^T Z_P$.

4. Naive Reckless Heuristic: We attempt to located baserunners that run often, but with low success rates, and vice versa. We set $g_4(P) = (Y_P{}^*)^T (1 - Z_P)$.

It is likely that more interesting heuristics exist, but these serve our purposes of conceptual simplicity.

3.3.3 Optimization Results

Now, for each heuristic g_i, we construct the MLB Burners (Cloggers) by taking the nine players with the best (worst) marks for g_i, performing the greedy algorithm to fix a batting order, and running the simulation to find X_T[10].

The Naive Aggressive heuristic, whose results are shown in Table 3, chose four catchers for the MLB Cloggers, and the team lost about 55 runs due to baserunning. Already, we have more than doubled the lower bound for X_T achieved by the White Sox. Similarly, the MLB Burners added about 41 runs due to baserunning, which was about twice as many as the our previous upper bound, achieved by the Mets. Note, however, that while the MLB Cloggers is composed almost entirely of older, slower players, the MLB Burners contains both players who are typically considered fleet afoot (Gathright, Taveras), and at least one (Uggla) who is not thought of as being much of a runner at all[11].

The Naive Conservative Heuristic selected those baserunners who are most (least) successful when they choose to run, regardless of how often they make that choice. Not surprisingly, the Burners include several players with high stolen base success rates, though not necessarily high stolen base totals. In particular, Carlos

[10] In all cases, we are looking at players with at least 500 opportunities to advance between 2005-2007.

[11] Indeed, Uggla has been successful in just 13 of 25 stolen base attempts in his career.

MLB Burners	$g_1(P)$	MLB Cloggers	$g_1(P)$
Gathright, Joey	1.48	Molina, Bengie	0.39
McDonald, John	1.47	Piazza, Mike	0.45
Gross, Gabe	1.45	Bard, Josh	0.54
Uggla, Dan	1.44	Mueller, Bill	0.54
Taveras, Willy	1.44	Thomas, Frank	0.55
Ozuna, Pablo	1.43	Saenz, Olmedo	0.64
Barmes, Clint	1.43	Garko, Ryan	0.65
Harris, Willie	1.42	Gibbons, Jay	0.65
Erstad, Darin	1.37	Hall, Toby	0.67
95% CI for X_T	41.0 ± 5.4	95% CI for X_T	-54.6 ± 6.2

Table 3: Optimization using g_1, the Naive Aggressive Heuristic

Beltran is usually credited with having the highest stolen base success rate of any player since caught stealings became an official statistic. Only one player on the Burners, Kevin Mench, stands out as a subpar runner. But while Mench does not run often, he rarely gets thrown out. The Cloggers, on the other hand, are populated almost entirely by catchers. Under this heuristic, the Burners gained almost 40 runs per season, while the Cloggers lost about 35 runs.

MLB Burners	$g_2(P)$	MLB Cloggers	$g_2(P)$
Matsui, Kazuo	2.72	Phillips, Jason	1.58
Drew, Stephen	2.64	Hall, Toby	1.83
Duffy, Chris	2.62	Lopez, Javy	1.87
Tyner, Jason	2.59	Ward, Daryle	1.87
Beltran, Carlos	2.59	Millar, Kevin	1.88
Mench, Kevin	2.58	Piazza, Mike	1.88
Infante, Omar	2.57	Kotchman, Casey	1.89
Damon, Johnny	2.57	Zaun, Gregg	1.92
Chavez, Endy	2.57	Miller, Damian	1.94
95% CI for X_T	39.4 ± 5.8	95% CI for X_T	-35.1 ± 5.8

Table 4: Optimization using g_2, the Naive Conservative Heuristic

The Naive Mixed heuristic identifies players who run both often and successfully, and it is not surprising that these players pushed the greatest lower bound on X to almost 70 runs per season, which is about 2.5 times what the Mets gained. Almost every player on the Burners is reknowned for his speed, with Gathright, Taveras, and Reyes ubiquitous candidates among those who are debated to be the

DOI: 10.2202/1559-0410.1174

game's fastest player[12]. On the other hand, the Cloggers consist of seven catchers and two first basemen / designated hitters. They lost about 43 runs due to baserunning, but fell short of the lower bound set by the Naive Aggressive Cloggers.

MLB Burners	$g_3(P)$	MLB Cloggers	$g_3(P)$
Gathright, Joey	1.84	Molina, Bengie	0.69
Taveras, Willy	1.81	Piazza, Mike	0.84
Reyes, Jose	1.78	Phillips, Jason	0.89
Harris, Willie	1.77	Hall, Toby	0.92
Figgins, Chone	1.76	Thomas, Frank	0.92
Logan, Nook	1.74	Garko, Ryan	0.96
Barfield, Josh	1.74	Bard, Josh	0.96
Ozuna, Pablo	1.73	Johjima, Kenji	1.01
Pierre, Juan	1.73	Treanor, Matt	1.02
95% CI for X_T	68.4 ± 5.5	95% CI for X_T	-42.5 ± 5.9

Table 5: Optimization using g_3, the Naive Mixed Heuristic

The Reckless heuristic was designed to select players for the Cloggers who ran often, but with low success rates, and vice versa for the Burners. It is interesting to note that this heuristic picked several players for the Burners who had been previously selected by different heuristics for different reasons (Matsui and Mench were on the Conservative Burners, while Mueller was on the Aggressive Cloggers). However, neither the Burners and the Cloggers demonstrated a statistically significant difference at the 5% level.

A summary of the results of the heuristic algorithms is shown in Table 7. The best performing heuristic was g_3, the Naive Mixed heuristic, which posted the highest value of X by a statistically significant margin, as well as a very low value of X_T. The lowest value of X_T was obtained from g_1, the Naive Aggressive heuristic.

4 Conclusion

4.1 Summary of Results

The results of our simulation generally confirm the estimates of Click and Fox, in that the impact of baserunning upon a major league team's run scoring ability is typically less than ± 25 runs per season. However, we show that even naive attempts

[12] Stark [11] listed Taveras, Reyes, and Pierre as the three fastest players in the National League, while Logan and Gathright were the two fastest in the American League, and Figgins was an honorable mention.

MLB Burners	$g_4(P)$	MLB Cloggers	$g_4(P)$
Mueller, Bill	0.32	Phillips, Jason	0.93
White, Rondell	0.34	Norton, Greg	0.86
Matsui, Kazuo	0.37	Sullivan, Cory	0.85
Mench, Kevin	0.37	Pena Jr., Tony	0.82
Quinlan, Robb	0.38	Hall, Toby	0.81
Duncan, Chris	0.38	Ellison, Jason	0.81
Luna, Hector	0.38	Gomez, Chris	0.81
Floyd, Cliff	0.38	Gonzalez, Luis	0.79
Bay, Jason	0.38	Cedeno, Ronny	0.78
95% CI for X_T	4.4 ± 6.2	95% CI for X_T	0.8 ± 5.2

Table 6: Optimization using g_4, the Naive Reckless Heuristic

g_i	Heuristic	X_T (Burners)	X_T (Cloggers)
g_1	Naive Aggressive	41.0 ± 5.4	-54.6 ± 6.2
g_2	Naive Conservative	39.4 ± 5.8	-35.1 ± 5.8
g_3	Naive Mixed	68.4 ± 5.5	-42.5 ± 5.9
g_4	Naive Reckless	4.4 ± 6.2	0.8 ± 5.2

Table 7: Summary of Results for Heuristic Algorithms

to assemble a team of the best (or worst) basreunners via simple heuristic algorithms can push this bound to at least 55, and as many as 70, runs per season. Lastly, our heuristics suggest that the best baserunners are those who run often and with high success rates, while the worst are those who don't try to advance often.

4.2 Improvements

While we believe that the model we have constructed is versatile and could be used to explore a number of different problems, the most obvious improvement would be to make the model more accurate by removing the slight under bias we identified in Section 3.1. The most natural way to do this would be to make the baserunning rules more realistic. Since runners are more likely to be moving with the pitch with two outs, a revision of the baserunning probabilities to take the number of outs into account could be fruitful.

Moreover, it is clear that the heuristics employed in this study were only the most naive and obvious, and a careful investigation would almost certainly yield better algorithms. These would be immediately meaningful to sabermetricians of all feathers.

DOI: 10.2202/1559-0410.1174

Lastly, it remains to explore the relationship between baserunning ability, batting ability, and batting order, which was largely ignored in this study. An investigation of this sort would be more complicated, but could yield interesting new results.

4.3 A Note on Openness

The code for this simulation was written in Java, using the open-source Eclipse IDE. The author is a firm believer in openness, and plans to make the source code for the simulation available as an open-source software development project. Unfortunately, the fundamental use of a proprietary statistical database makes this impossible at the present time. Nevertheless, the source code could be modified to access an open data source, such as Retrosheet, instead of the currently queried proprietary database. As development continues on the RetroSQL project, we hope to open this project to the public. Anyone interested in assisting in this process is encouraged to contact the author.

References

[1] J. Albert. *Teaching Statistics Using Baseball*. Maa, 2003.

[2] J. Albert and J. Bennett. *Curve Ball: Baseball, Statistics, and the Role of Chance in the Game*. Copernicus Books, 2003.

[3] B. Bukiet, E.R. Harold, and J.L. Palacios. A Markov Chain Approach to Baseball. *Operations Research*, 45(1):14–23, 1997.

[4] J. Click. Station to Station: The Expensive Art of Baserunning. In *Baseball Prospectus 2005*, pages 511–519. Workman Publishing (New York), 2005.

[5] E. Cook. *Percentage Baseball*. Waverly Press, Inc., (Baltimore), 1964.

[6] D. Fox. Circle the Wagons: Running the Bases Part III. url: http://www.hardballtimes.com/main/article/circle-the-wagons-running-the-bases-part-iii/, 2005.

[7] R.A. Freeze. An Analysis of Baseball Batting Order by Monte Carlo Simulation. *Operations Research*, 22(4):728–735, 1974.

[8] M. Lewis. *Moneyball: The Art of Winning an Unfair Game*. WW Norton & Company, 2004.

[9] M.D. Pankin. Evaluating Offensive Performance in Baseball. *Operations Research*, 26(4):610–619, 1978.

[10] J.S. Sokol. A Robust Heuristic for Batting Order Optimization Under Uncertainty. *Journal of Heuristics*, 9(4):353–370, 2003.

[11] J Stark. Who's the fastest? url: http://sports.espn.go.com/mlb/columns/story?columnist=stark_jayson&id=2137637, 2005.

[12] J. Thorn and P. Palmer. *The hidden game of baseball: a revolutionary approach to baseball and its statistics*. Doubleday, Garden City, NY, 1984.

[13] T.L. Turocy. Offensive performance, omitted variables, and the value of speed in baseball. *Economics Letters*, 89(3):283–286, 2005.

DOI: 10.2202/1559-0410.1174