Optimal Strategy to Hold Ronald Acuña Jr. On

Harrison Stanton

December 5, 2023

1 Introduction

Prior to the 2023 MLB season, holding a runner on could be modeled as an infinitely repeated game of imperfect information. A pitcher could attempt to pick a runner off as many times as he wanted to. However, in order to speed up the pace of play for the 2023 season, MLB added a rule saying that pitchers were allowed two disengagements per at-bat. If they disengaged the rubber a third time and did not get the runner out, then the runner was awarded the next base. This drastically changes how the pickoff scenario is modeled. We explore the effects of the new pickoff rule in this paper.

2 The Model

2.1 Players and Strategies

After the rule changes, holding a runner on can now be modeled as a finitely repeated game of imperfect information with three periods, with each disengagement from the rubber resulting in an advancement of a period. There are two players in this game: the pitcher (P) who can pickoff (p) or not (p') and the runner (R) who can steal (s) or not (s'). In this model, the players move simultaneously. Each iteration of the game can be modeled in the normal form, as shown below, where the first number in each ordered pair is the payoff for the pitcher and the second number is the payoff for the runner:

	Runner: s	Runner: s'
Pitcher: p	(a, b)	(c, d)
Pitcher: p'	(e, f)	(g, h)

2.2 Assumptions of the Model

There are a couple of key assumptions made when modeling the pickoff scenario. The first assumption is that if a runner is stealing and the pitcher picks off, then the runner is out. The second assumption is that if the pitcher picks off and the runner is not stealing, the runner is safe. Neither of these assumptions is true 100% of the time, but both are close enough to true that we learn important information from the model. Another assumption made in this model is that the decision of the runner to steal and the pitcher to pickoff are made simultaneously, and that the information is imperfect. This assumption fails to hold in the instance of a lefty pitcher who has a read pickoff move, but otherwise, with a righty pitcher or a lefty pitcher with a poor pickoff move, this assumption holds.

3 Payoff Numbers

In order to solve the model, there need to be payoff numbers that accurately reflect the payoff for each combination of events. In order to calculate the payoff numbers, a couple of key components were used. The first is the Expected Runs (ER) matrix[1 2]. The ER matrix for the 2022 season was used to represent the expected number of runs a team would score in the rest of the inning when at each state. 2022 data was used because the 2023 play-by-play data had not been released on Retrosheet at the time of this project. This also meant there was no way to generalize the model for all baserunners and states. This meant the most effective way to study the effects of the new rule was to observe one specific baserunner at one specific state. In 2023, Ronald Acuña Jr had the most stolen base attempts, so his attempts were observed. The state of nobody out with Acuña Jr. on first was chosen to be the state observed. In order to obtain his stolen-base percentage only from first to second base, anytime Acuña Jr. stole third was filtered out of his stolen base numbers.

3.1 Calculating Payoff Numbers

Acuña Jr.'s stolen-base percentage from 2023 after filtering out times where he stole third is 88.1%. From here, the expected run value[3] for each combination of strategies ($\{p, s\}, \{p', s\}, \{p, s'\}, \{p', s'\}$) needed to be calculated for the pitcher and Acuña Jr. This is a zero-sum game, so the expected run value for the pitcher will always be the negative expected run value for Acuña Jr. In the situation where the pitcher plays p' and the Acuña Jr. plays s', the expected run value is 0 for both players because they remain in the same state. If the pitcher plays p and the runner plays s, then the payoff is as follows[4]:

$$\mathrm{P(safe)} * ((100, 0 \text{ outs}) - (010, 0 \text{ outs})) + \mathrm{P(out)} * ((100, 0 \text{ outs}) - (000, 1 \text{ out}))$$

$$= 1 * (.87 - .25)$$

This calculation tells us the pitcher's expected run value is 0.62 while Acuña Jr's is -0.62. This same process was repeated for the other two strategies. For

[1]The ER matrix was made in R, and the code is attached at the end of the paper

[2]Help from source: Marchi, Max, et al. Analyzing Baseball Data with R. 2nd ed., CRC Press, 2019.

[3]Statcast's official definition for run value includes the ball and strike count in their ER matrix: these were not factored into this study

[4](100, 0 outs) refers to row and column of the ER matrix in Table 1 of the Appendix

$\{p', s\}$, Acuña Jr.'s expected run value is obtained by solving:

$$\text{P(safe)} * ((010, 0 \text{ outs}) - (100, 0 \text{ outs})) + \text{P(out)} * ((000, 1 \text{ out}) - (100, 0 \text{ out}))$$

$$= (.881)(1.06 - .87) + (.119)(.25 - .87) = .09$$

and the pitcher's expected run value is -.09. The last payoff number to solve for is the expected run value in the last period when $\{p, s'\}$ is played. Because one of the assumptions is that if Acuña Jr. isn't stealing and the pitcher picks off he is safe every time, then the expected run value is calculated by:

$$(010, 0 \text{ outs}) - (100, 0 \text{ outs})$$

$$= 1.06 - .87 = .19$$

This means Acuña's expected run value is .19 and the pitcher's is -.19.

4 Solving the model

In solving this game, we are going to look for Subgame Perfect Nash Equilibria (SPNE). There will be no pure-strategy SPNE in this model, because the pitcher and the runner have competing incentives. This means that we will be looking for mixed-strategy SPNE. The first step to doing this is looking in the last branch of our extensive form model, as shown in Figure 1.

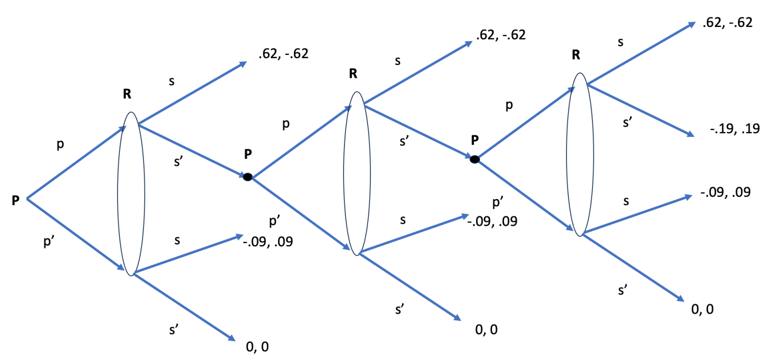

Figure 1: Full Extensive Form

In order to solve for a mixed-strategy Nash Equilibrium, we need to solve for each player's indifference conditions. This can be calculated for the pitcher as follows:

$$\mathbb{E}\Pi_{\text{P}}(p) = \mathbb{E}\Pi_{\text{P}}(p')$$

$$0.62s + -0.19s' = -0.09s + 0s'$$

We solve this out, and find that the pitcher is indifferent in the last stage of the game when the runner steals 21% of the time, or in other words, when $s = 0.21$. Now, this process is repeated for the runner.

$$\mathbb{E}\Pi_{\mathrm{R}}(s) = \mathbb{E}\Pi_{\mathrm{R}}(s')$$

$$-0.62p + 0.09p' = 0.19p + 0p'$$

Solving this equation, the runner's indifference condition occurs when the pitcher picks off 10% of the time ($p = 0.1$). Because of how the model is set up, with a simultaneous decision where each player does not observe the other's actions, this means that the events are independent of one another. Therefore, the probability the pitcher picks and the runner steals is the probability that one happens times the probability the other happens. Using the p and s values found in the step above and knowing they are independent, the expected run value for Acuña Jr. can be calculated as follows:

$$-.62\mathrm{P}(p \cap s) + .19\mathrm{P}(p \cap s') + .09\mathrm{P}(p' \cap s) + 0\mathrm{P}(p' \cap s')$$

$$= -.62(.1)(.21) + .19(.1)(.79) + .09(.9)(.21) = .019$$

And the pitcher's expected run value is -.019. From here, the last branch is simplified to the ordered pair (-.019, .019), as shown in Figure 2. The same

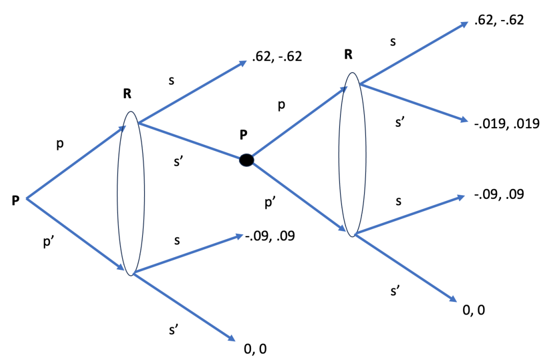

Figure 2: Extensive Form with last branch simplified

steps are repeated here, solving the equations

$$\mathbb{E}\Pi_{\mathrm{P}}(p) = \mathbb{E}\Pi_{\mathrm{P}}(p')$$

and

$$\mathbb{E}\Pi_{\mathrm{R}}(s) = \mathbb{E}\Pi_{\mathrm{R}}(s')$$

for the pitcher's and runner's indifference conditions. In stage 2, the pitcher is indifferent when the runner plays $s = .026$ and the runner is indifferent when the pitcher plays $p = .123$. The expected run value for Acuña Jr. can be calculated in the same way as in the last branch, by solving the equation below:

$$-.62\mathrm{P}(p \cap s) + .019\mathrm{P}(p \cap s') + .09\mathrm{P}(p' \cap s) + 0P(p' \cap s')$$

$$-.62(.123)(.026) + .019(.123)(.974) + .09(.877)(.026) = .0023$$

while the pitcher's is $-.0023$. The ordered pair $(-.0023, .0023)$ is replaced in our extensive form diagram, leading to Figure 3.

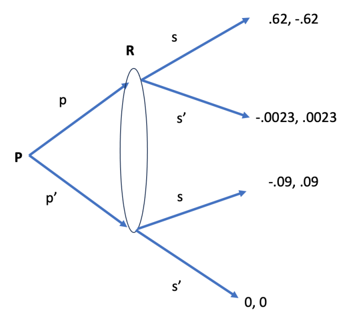

Figure 3: Fully simplified Extensive Form

We repeat this process again for the simplified extensive form model, solving for the pitcher and runner indifference conditions by setting the expected values of either strategy for the player equal to each other. The pitcher is indifferent when the runner plays s = .003, and the runner is indifferent when the pitcher plays p = .126, which results in an expected run value of .00028 for Acuña Jr. and -.00028 for the pitcher. This leads us to a mixed strategy SPNE when the pitcher plays $p = \{0.126, 0.124, 0.1\}$ and the runner plays $s = \{0.003, 0.026, 0.21\}$. The first term in the pitcher's strategy space means that he should pickoff 12.6% of the time if he has not yet picked off during the at-bat, 12.4% when he has picked off once, and 10% of the time when he has already picked off twice in an at-bat. Similarly, Acuña Jr.'s strategy space means that .3% of the time he should steal when the pitcher has not picked off during the at-bat, he should steal 2.6% of the time when the pitcher has already picked off once, and steal 21% of the time when the pitcher has already picked off twice. In this scenario, both the pitcher and Acuña Jr. are playing a best response to the other persons actions.

5 Results

5.1 Analysis of the Equilibrium

The SPNE makes logical sense in this game. The optimal outcomes for the pitcher are either he picks Acuña Jr. off, or the pitcher does not pick and Acuña

Jr. does not attempt to steal at all. The only way for these two outcomes to happen are if he mixes between p and p'. Since Acuña Jr. is going to be playing s' more often than s, the pitcher wants to play p' more often than he plays p. The same thing goes for Acuña Jr. His best outcomes happen when he steals and the pitcher does not pickoff, or when the pitcher picks off and he does not steal. This means he is going to mix between stealing and not stealing. Additionally, since the cost of being picked off far outweighs the gain of getting to the next base, the rate at which he attempts to steal second base needs to stay fairly low, as seen in our equilibrium. One insightful piece of information in the SPNE is in the third time period, or after the pitcher has already attempted 2 pickoffs. In our SPNE, the pitcher is playing $p = .1$ and Acuña Jr. is playing $s = .21$. This is a little counter-intuitive that the pitcher's pickoff rate would remain this high in the third time period. Throughout the season in 2023, pitchers seemed hesitant to pickoff in the third time period because of the new disengagement rule and the threat of giving up the next base. However, since the runner's steal rate is going to be higher in the third period, the pitcher is actually going to want to pickoff nearly as often as he did in the other periods in hopes of gaining the 0.62 expected run value. Because a Nash Equilibrium means that both players are playing a best response to the other, the conclusion that can then be drawn is that in 2023, either the pitcher, Acuña Jr., or both were not playing best responses to the other's actions. This means that they were potentially leaving runs on the table.

5.2 Limitations

There are some clear limitations to this model. The first is that the model is only representing when Acuña Jr. is on first base with no outs. In order to gain insight on the optimal strategies to hold Acuña Jr. on in other situations, the payoff numbers from the ER matrix would have to be substituted into the payoff number calculations. Another limitation of the model is that the 2023 data was not available, and because the new disengagements rule could affect how runs are scored, there might be differences in the 2023 ER matrix. The third limitation of the model is that it works better to model a righty pitcher holding Acuña Jr. on because a righty cannot observe whether or not Acuña Jr. is stealing the way that a lefty pitcher can.

5.3 Further Areas of Study

In the future, it would be interesting to see how or if Acuña Jr.'s stolen-base percentage changes after each pickoff attempt. During the research, his stolen-base percentage after each number of pickoff attempts was observed. However, there were not enough observations to make any definitive claims as to whether or not the stolen-base percentage was different after each pickoff attempt (95% confidence intervals were made to test the stolen-base percentages after 0 picks, 1 pick, and 2 picks, and the intervals for after 0 and 1 picks both contained numbers greater than one. Additionally, the confidence interval for after 2 picks ranged from about .1 to .9, so they were not super insightful). This could significantly change the optimal strategy if Acuña Jr.'s stolen-base percentage increased or decreased over time. For example, in the sample data, his stolen-base percentage was actually about 6 percent higher if the pitcher picked 0 times versus if the pitcher picked 1 time. Once there are more observations, it would be interesting to see if this trend holds, which would significantly affect how often Acuña Jr. steals in each stage of the game.

Another extension of this project that could be insightful is to model the situation where a lefty is holding Ronald Acuña Jr on. Then, instead of a model of imperfect information, this game could be modeled so that the pitcher gets to observe Acuña Jr.'s move, and react accordingly.

6 Appendix

	0 outs	**1 out**	**2 outs**
000	0.47	0.25	0.10
001	1.27	0.96	0.38
010	1.06	0.67	0.31
011	1.98	1.41	0.55
100	0.87	0.51	0.21
101	1.75	1.16	0.51
110	1.44	0.90	0.43
111	2.40	1.53	0.77

Table 1: ER Matrix, 2022 MLB Season

ER matrix project

2023-11-17

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("utils")
```

##reading in CSV

```
data2022 <- read.csv("Retrosheets_for_ERmatrix.csv")
```

##creating runs, half inning, and runs scored variables

```
data2022 %>%
  mutate(RUNS = vis_score + home_score,
         HALF.INNING = paste(game_id, inning, batting_team),
         RUNS.SCORED =
         (batt_dest > 3) + (run_1b_dest > 3) + (run_2b_dest > 3) + (run_3b_dest > 3)) ->
  data2022
```

##creating half inning data frame

```
data2022 %>%
  group_by(HALF.INNING) %>%
  summarize(outs.inning = sum(event_outs),
            Runs.Inning = sum(RUNS.SCORED),
            Runs.Start = first(RUNS),
            MAX.RUNS = Runs.Inning + Runs.Start) ->
half_innings
```

##inner joining the half inning and data 2022 data frames

```
data2022 %>%
  inner_join(half_innings, by = "HALF.INNING") %>%
  mutate(RUNS.ROI = MAX.RUNS - RUNS) ->
  data2022
```

##Creating the state variable

```
data2022 %>%
  mutate(BASES =
           paste(ifelse(run_1b > '', 1, 0),
             ifelse(run_2b > '', 1, 0),
```

```
             ifelse(run_3b > '', 1, 0), sep = ''),
             STATE = paste(BASES, outs)) ->
  data2022
```

##Creating NRunner variable, tells when state changes

```
data2022 %>%
  mutate(NRUNNER1 =
           as.numeric(run_1b_dest == 1 | batt_dest == 1),
         NRUNNER2 =
           as.numeric(run_1b_dest == 2 | run_2b_dest == 2 | batt_dest == 2),
         NRUNNER3 =
           as.numeric(run_1b_dest == 3 | run_2b_dest == 3 | run_3b_dest == 3 | batt_dest == 3),
         NOUTS = outs + event_outs,
         NEW.BASES = paste(NRUNNER1, NRUNNER2, NRUNNER3, sep = ""),
         NEW.STATE = paste(NEW.BASES, NOUTS)) ->
  data2022
```

##restricting to plays with state change only

```
data2022 %>%
  filter((STATE != NEW.STATE) | (RUNS.SCORED > 0)) ->
  data2022
```

##filtering out innings that ended in a walkoff (less than 3 outs)

```
data2022 %>%
  filter(outs.inning == 3) -> data2022C
```

##averaging all the states

```
data2022C %>%
  group_by(STATE) %>%
  summarize(Mean = mean(RUNS.ROI)) %>%
  mutate(Outs = substr(STATE, 5, 5)) %>%
  arrange(Outs) -> RUNS
```

##making the data frame into a matrix

```
RUNS_out <- matrix(round(RUNS$Mean, 2), 8, 3)
dimnames(RUNS_out)[[2]] <- c("0 outs", "1 out", "2 outs")
dimnames(RUNS_out)[[1]] <- c("000", "001", "010", "011", "100", "101", "110", "111")
```

##displaying the matrix

```
print(RUNS_out)
```

```
##     0 outs 1 out 2 outs
## 000   0.47  0.25   0.10
## 001   1.27  0.96   0.38
## 010   1.06  0.67   0.31
## 011   1.98  1.41   0.55
## 100   0.87  0.51   0.21
## 101   1.75  1.16   0.51
## 110   1.44  0.90   0.43
## 111   2.40  1.53   0.77
```