

# *DSSR*

Discussion Paper No. 24

**On Markov perfect equilibria in baseball**

Akifumi Kira  
Keisuke Inakawa

March, 2014

Data Science and Service Research  
Discussion Paper

---

Center for Data Science and Service Research  
Graduate School of Economic and Management  
Tohoku University  
27-1 Kawauchi, Aobaku  
Sendai 980-8576, JAPAN

# On Markov perfect equilibria in baseball

Akifumi KIRA

Graduate School of Economics and Management  
Tohoku University  
a-kira@econ.tohoku.ac.jp

Keisuke INAKAWA

Faculty of Systems Science and Technology  
Akita Prefectural University  
inakawa@akita-pu.ac.jp

## Abstract

We formulate baseball as a finite Markov game with approximately 3.5 million states. The manager of each opposing team is the player who maximizes the probability of their team winning. We derive, using dynamic programming, a recursive formula which is satisfied by Markov perfect equilibria and the value functions of the game for both teams. By solving this recursive formula, we can obtain optimal strategies for each condition. We demonstrate with numerical experiments that these can be calculated in approximately 1 second per game.

**Keywords** Markov game, dynamic programming, optimal strategies in baseball.

## 1 Introduction

Similar to the various problems within corporations, even in sports, there are several successful cases that have employed scientific methods. For example, using a technique known as Sabermetrics, Billy Beane, the general manager of Oakland Athletics in the Major League, built up a powerful team on a low budget. And in Japanese professional baseball, making full use of the information system BOS (Baseball Operation System) contributed to the successes of the Hokkaido Nippon-Ham Fighters, who won the league four times, and became the top team in Japan once in the seven years after 2006.

On the other hand, a method exists for analyzing baseball that uses Markov chains, and has been extensively studied by several authors (Bellman [1], Bukiet et al. [2], Cover and Keilers [3], Howard [5], Turocy [8], and others). The first scholar to have applied Markov chains to Baseball, seems to be Howard [5]. He said that the discrete nature of events in baseball constituted a natural Markov model. He suggests the optimization of strategies (especially sacrifice bunts and stolen bases), which maximizes the expected runs scored in one inning by using dynamic programming. In other words, this is a Markov decision process (MDP) approach. Bellman [1] proposes a more detailed formulation. His model analyzes not only a batter-by-batter level, but also a pitch-by-pitch level. He provided a subtle insight into strategies through a discussion based on the two criteria of maximizing the expected number of runs scored, as well as the threshold probability of scoring at least  $k$  runs in one inning. However, due to the shortage of computing capacity at the time, these approaches using dynamic programming were not implemented.

At the same time as the work of Bellman, in 1977, D'Esopo and Lefkowitz [4] propose a scoring index (SI) as an evaluation index for the expected number of runs scored in one inning assuming that the same player steps up to the plate repeatedly. Under the same assumption,

Cover and Keilers [3] in 1977 propose a similar index, the OERA (Offensive Earned-Run Average) value, as an index to evaluate the expected number of runs scored in a single game. In the OERA model, the baseball rules are simplified to apply the absorbing Markov chain model in the calculation of the expected runs scored. Bukiet et al. [2] in 1997 take into consideration of the batting order. They give an algorithm to calculate the expected runs scored in a game assuming the nine players constituting the team step up to the plate in a given order. Further, a method for approximately computing the winning percentage is also proposed. These studies can be classified as Markov chain approaches with matrix analysis.

In 2008, Turocy [8] produces a model to also couple the strategies of the opposing team with the Bellman’s approach by using a Markov game, which is a multi-agent extension of MDPs (e.g. see Shapley [7] and Zachrisson [9]). Here, the manager of each opposing team maximizes the probability of their team winning. Since the model adopts the MLB rule to play extra innings until a winner is determined, the states for an extra inning can be identified as being the same as for the ninth inning. In addition, by establishing an upper limit of 30 runs for the run difference (mercy-rule), his stochastic game has a finite number of states, and a finite number of actions, and so a Markov perfect equilibrium (MPE) exists. The total number of states in this case is approximately 2.13 million. Turocy performed numerical experiments using backward induction (dynamic programming) from the start of the game up to the completion of the eighth inning, and using a fixed-point approximation by a value-function iteration for the ninth inning. The details of such as the recursive formula and the algorithm are omitted in the paper, but he states that the values of the game (the equilibrium winning percentages for both teams) could be solved with high accuracy in less than a minute.

In this paper, similar to Turocy, we consider a game-theoretic approach for baseball. Baseball is formulated as a finite Markov game with approximately 3.5 million states. Here, whereas Turocy counts the combination of all the states including those that are unreachable, in our model, only the states which are reachable are counted. We also suppose that the manager of each team maximizes the probability of their team winning. The principal differences compared to Turocy’s model are the two points that, (a) in order to consider the success rate with stolen bases in more detail, the identity of the runner on first base is stored in the state, and (b) the rule from Japanese baseball that extra innings are restricted to a maximum of three is included, so the game may end in a draw. Because of the mercy-rule and the finiteness of the number of extra innings, the length of the game also becomes finite. Hence, at least one pure-strategy MPE does exist in our model, and they can be found using backward induction. We derive a recursive formula that is satisfied by the MPEs and the value functions of the game, and carry out numerical experiments to solve it.

In Section 2, we provide a finite Markov game as a formulation for baseball. States, actions, state transitions with simplifying rules, and payoff functions are all defined in detail. Here, we make use of the MDP methodology, which reduces a probability criterion down to an expectation criterion (e.g. Kira et al. [6]). In Section 3, we define MPEs and the value functions of the game, and derive the recursive formula that is satisfied by them. By solving this, the optimal strategies for each state are obtained, such as a sacrifice bunt or a stolen base. Section 4 performs computational experiments. We calculate the recursive formula for just reachable states from the initial state by implementing the depth-first search of the game tree. In this way, the value functions of the game for both teams and a pure-strategy MPE can be found in about 1 second. The results obtained from numerical examples in this study are described. Concluding remarks and future direction are discussed in Section 5.

## 2 Formulation as a Markov game

In this section, baseball is formulated as a Markov game.

## 2.1 States

Let  $\mathcal{S}$  be the state space. A state  $s = (\iota, \tau, \omega, \lambda, \mathbf{r}, \mathbf{b}) \in \mathcal{S}$  is made up of 6 components. Each component is defined as follows:

1.  $\iota \in \{1, 2, \dots, 12\}$  represents the current inning.  $\iota = 9$  is the final inning, and for a tie, extra innings are played up to a maximum of  $\iota = 12$ .
2.  $\tau \in \{0, 1\}$  represents offense in the top half of the inning ( $\tau = 0$ ) or offense in the bottom half of the inning ( $\tau = 1$ ).
3.  $\omega \in \{0, 1, 2, 3\}$  represents the current number of outs.
4.  $\lambda$  is the current run difference and represents the value found by subtracting the runs of the home team (last-batting team) from the runs of the away team (first-batting team).
5.  $\mathbf{r} = (r_3, r_2, r_1)$  represents the state of the runners.
  - $r_3 \in \{0, 1\}$  takes a value 0 if there is no runner on third base, and a value 1 if a runner is present.
  - $r_2 \in \{0, 1\}$  takes a value 0 if there is no runner on second base, and a value 1 if a runner is present.
  - $r_1 \in \{0, 1, \dots, 9\}$  takes a value 0 if there is no runner on first base, and the same value as the batting order of the runner if a runner is present.

Only  $r_1$  distinguishing between runners is in order to take into account the success rate, which is dependent on the runner when performing a stolen base from first to second base. In this paper, neither a stolen base from second to third base, nor a stolen base from third to home base, are considered.

6.  $\mathbf{b} = (b_1, b_2)$  indicates to which batter the batting order rotates.
  - $b_0 \in \{1, 2, \dots, 9\}$  indicates to which batter the batting order of the first-batting team rotates. It represents, when in offense ( $\tau = 0$ ), that the  $b_0$ -th batter steps up to the plate. When in defense ( $\tau = 1$ ), it means that the leadoff hitter in the next inning is the  $b_0$ -th batter.
  - $b_1 \in \{1, 2, \dots, 9\}$  similarly indicates to which batter the batting order of the last-batting team rotates.

The initial state  $s_0$  at the start time of the game is as follows:

$$s_0 = (\iota, \tau, \omega, \lambda, \mathbf{r}, \mathbf{b})_0 = (1, 0, 0, 0, (0, 0, 0), (1, 1)).$$

$\mathcal{S}_Q$  denotes the total states (absorbing states) at the end of the game:

$$\mathcal{S}_Q := \mathcal{S}_Q^0 \cup \mathcal{S}_Q^1 \cup \mathcal{S}_Q^2 \cup \mathcal{S}_Q^3,$$

where

$$\begin{aligned} \mathcal{S}_Q^0 &= \{s \in \mathcal{S} \mid \iota \geq 9, \tau = 1, \omega = 3, \lambda > 0\}, \\ \mathcal{S}_Q^1 &= \{s \in \mathcal{S} \mid \iota = 9, \tau = 0, \omega = 3, \lambda < 0\} \cup \{s \in \mathcal{S} \mid \iota \geq 9, \tau = 1, \lambda < 0\}, \\ \mathcal{S}_Q^2 &= \{s \in \mathcal{S} \mid \iota = 12, \tau = 1, \omega = 3, \lambda = 0\}, \\ \mathcal{S}_Q^3 &= \{s \in \mathcal{S} \mid \lambda \leq -30\} \cup \{s \in \mathcal{S} \mid \lambda \geq 30\}. \quad (\text{mercy-rule}) \end{aligned}$$

$\mathcal{S}_Q^0$  and  $\mathcal{S}_Q^1$  correspond to a victory for the first-batting team and a victory for the last-batting team, respectively.  $\mathcal{S}_Q^2$  corresponds to a tie in the 12th inning after playing extra innings. We

note that the mercy-rule refers to the establishment of a called game during the inning. By adopting both the Japanese professional baseball rules (i.e., the number of extra innings is finite) and the above mercy-rule, we get a finite Markov game. In addition, we equate the state  $s$ , such that  $\omega = 3$  and  $s \notin \mathcal{S}_Q$ , with the corresponding state after the inning is over.

## 2.2 Actions

The manager of each team is the player maximizing the probability of their team winning. For each state  $s \in \mathcal{S}$ , if  $\tau = 0$  (the top half of the inning) then the first-batting team can choose their actions. Conversely, if  $\tau = 1$  (the bottom half of the inning) then the last-batting team can choose. Here, we let  $\mathcal{S}_0$  and  $\mathcal{S}_1$  be the sets of states of moves for, respectively, the first-batting team and the last-batting team. Namely,

$$\begin{aligned} \mathcal{S}_0 &= \{s \in \mathcal{S} \mid \tau = 0, s \notin \mathcal{S}_Q\}, \\ \mathcal{S}_1 &= \{s \in \mathcal{S} \mid \tau = 1, s \notin \mathcal{S}_Q\}. \end{aligned}$$

In this paper, the action space is defined as

$$\mathcal{A} = \{\text{batting, stolen base, sacrifice bunt}\}.$$

Let us consider a point-to-set valued mapping  $\mathcal{A} : \mathcal{S} \rightarrow 2^{\mathcal{A}} \setminus \{\emptyset\}$ .  $\mathcal{A}(s)$ , called the feasible action space, represents the set of all actions in state  $s$ . In this paper, we define them in the following manner:

$$\begin{aligned} \text{batting} &\in \mathcal{A}(s), \quad \forall s \in \mathcal{S}, \\ \text{stolen base} &\in \mathcal{A}(s) \iff r_2 = 0, \quad r_1 \geq 1, \\ \text{sacrifice bunt} &\in \mathcal{A}(s) \iff \omega = 0 \text{ or } 1, \quad r_3 + r_2 + r_1 \geq 1. \end{aligned}$$

Hence, stolen bases are feasible if and only if there is no second base runner and there is a runner present on first base, and sacrifice hits are feasible if and only if a runner is present with 0 or 1 outs.

**Definition 2.1** (Markov policy). *A mapping  $\pi_0 : \mathcal{S}_0 \rightarrow \mathcal{A}$  is called a (deterministic) Markov policy for the first-batting team if  $\pi_0(s) \in \mathcal{A}(s)$  for all  $s \in \mathcal{S}_0$ . Similarly, a mapping  $\pi_1 : \mathcal{S}_1 \rightarrow \mathcal{A}$  is called a Markov policy for the last-batting team if  $\pi_1(s) \in \mathcal{A}(s)$  for all  $s \in \mathcal{S}_1$ .*

We denote the set of all deterministic Markov policies for the first-batting team and the last-batting team by  $\Pi_0$  and  $\Pi_1$ , respectively.

## 2.3 State transitions

For any  $a \in \mathcal{A}$ , let us define  $\mathcal{X}(a)$ , the set of all results that can occur stochastically when the action  $a$  is chosen, as follows:

$$\mathcal{X}(a) = \begin{cases} \{\text{out, single, double, triple, home run, walk}\} & \text{if } a = \text{batting} \\ \{\text{success, fail}\} & \text{otherwise.} \end{cases}$$

We denote the graph of  $\mathcal{A}(\cdot)$  by  $G_r(\mathcal{A})$ . Namely,

$$G_r(\mathcal{A}) = \{(s, a) \mid a \in \mathcal{A}(s), s \in \mathcal{S}\}.$$

For any  $(s, a) \in G_r(\mathcal{A})$  and any  $x \in \mathcal{X}(a)$ ,  $p(x \mid s, a)$  represents the conditional probability with which the result  $x$  occurs, given that action  $a$  is chosen in state  $s$ .

$$\begin{aligned} p(\cdot \mid s, a) &: \mathcal{X}(a) \rightarrow [0, 1], \quad \forall (s, a) \in G_r(\mathcal{A}), \\ \sum_{x \in \mathcal{X}(a)} p(x \mid s, a) &= 1, \quad \forall (s, a) \in G_r(\mathcal{A}). \end{aligned}$$

Now, with the definition of the state, the state  $s = (\iota, \tau, \omega, \lambda, \mathbf{r}, \mathbf{b})$  includes information about the inning number, whether it is the top or bottom half, the out count, the run difference, the batting order of the batters, whether there are runners present or not, and also, the batting order of the runner if one is present on first base. Therefore, the transition probability generally depends on all of these states. However, in the numerical experiments carried out in Section 4, we assume that the transition probability depends only on the players that make a hit, or perform sacrifice hits or stolen bases, and does not depend on those other components which comprise the state.

This paper simplifies baseball in a similar manner to previous research. The simplifying rules used in this paper are as below.

“Simplifying rules”

1. With a mishit (an out), neither a batter nor a runner can advance bases.
2. A single advances a runner on first base to third base, and runners on second and third base reach the home plate.
3. A double and a triple allows all runners to reach the home plate.
4. It is assumed that there are no double plays.
5. For a successful stolen base, the runner on first base advances to second base.
6. For an unsuccessful stolen base, the runner on first base is out.
7. For a successful sacrifice hit, the runners advance one base forward, and the batter performing the sacrifice hit is out.
8. For an unsuccessful sacrifice hit, the runner closest to the home plate is out, the other runners advance one base forward, and the batter is then the runner on first base.

If these simplifying rules are followed, then the next state  $s'$  is determined uniquely when in state  $s$ , action  $a$  is chosen, and result  $x$  occurs. We denote this next state by

$$s' = t(s, a, x).$$

## 2.4 Payoff function

Since, for any Borel set  $\mathcal{B}$  and any random variable  $X$ , we know the relation

$$\Pr(X \text{ is in } \mathcal{B}) = \mathbb{E}[\mathbf{1}_{\mathcal{B}}(X)], \quad \mathbf{1}_{\mathcal{B}}(X) = \begin{cases} 1 & \text{if } X \text{ is in } \mathcal{B}, \\ 0 & \text{otherwise.} \end{cases}$$

The approach, using this relation to reduce a probability criterion to an expectation criterion, is often used in the field of Markov decision processes (see Kira et al. [6]). Thus, we define the terminal payoff functions  $r_0, r_1 : \mathcal{S}_Q \rightarrow \{0, 1\}$  for the last-batting team and the first-batting team as follows:

$$r_0(s) = \begin{cases} 1 & \lambda > 0, \\ 0 & \lambda \leq 0, \end{cases} \quad r_1(s) = \begin{cases} 0 & \lambda \geq 0, \\ 1 & \lambda < 0. \end{cases}$$

If the game is won, a payoff of 1 is acquired, whereas a loss or a tie is a payoff of 0. Suppose that Markov policies  $\pi_0 \in \Pi_0$  and  $\pi_1 \in \Pi_1$  are employed by the first-batting team and the last-batting team, respectively. In this case, the Markov game commencing from each state  $s$  can

be regarded as a Markov chain. In other words, if we let  $X_n$  be the state after  $n$  step transition from the initial state  $s_0$ , then  $\{X_n\}$  is the Markov chain satisfying

$$\Pr^{\pi_0, \pi_1}(X_{n+1} = s' | X_n = s) = \begin{cases} p(x | s, \pi_i(s)) & \text{if } s \in \mathcal{S}_i, s' = t(s, \pi_i(s), x), \\ 1 & \text{if } s \in \mathcal{S}_Q, s' = s, \\ 0 & \text{otherwise.} \end{cases}$$

Let  $T$  be the arrival time of  $\{X_n\}$  to  $\mathcal{S}_Q$ . Namely,

$$T := \min\{n | X_n \in \mathcal{S}_Q\} < \infty.$$

We can define the probabilities of the first-batting team winning and the last-batting team winning, denoted by  $v_0(s; \pi_0, \pi_1)$  and  $v_1(s; \pi_0, \pi_1)$  respectively, as follows:

$$\begin{aligned} v_0(s; \pi_0, \pi_1) &= \mathbb{E}^{\pi_0, \pi_1} [r_0(X_T) | X_0 = s], \\ v_1(s; \pi_0, \pi_1) &= \mathbb{E}^{\pi_0, \pi_1} [r_1(X_T) | X_0 = s]. \end{aligned}$$

### 3 Markov perfect equilibria and dynamic programming

In this section, we define a Markov perfect equilibrium and the value functions of the game, and derive the recursive formula for effectively computing them.

A Markov perfect equilibrium is a profile of Markov policies that yields a Nash equilibrium in every proper subgame.

**Definition 3.1** (Markov perfect equilibrium, MPE). *A profile of (deterministic) Markov policies  $(\pi_0^*, \pi_1^*)$  for the first-batting team and the last-batting team is called a (pure-strategy) MPE if it is a subgame perfect equilibrium. Namely, it satisfies*

$$\begin{aligned} v_0(s; \pi_0, \pi_1^*) &\leq v_0(s; \pi_0^*, \pi_1^*), \quad \forall s \in \mathcal{S}, \quad \forall \pi_0 \in \Pi_0, \\ v_1(s; \pi_0^*, \pi_1) &\leq v_1(s; \pi_0^*, \pi_1^*), \quad \forall s \in \mathcal{S}, \quad \forall \pi_1 \in \Pi_1. \end{aligned}$$

**Remark 3.1** (existence of the equilibria). *In the most general context, the action chosen by a policy in each state may be randomized. However, it is well-known that at least one pure-strategy MPE exists for a finite Markov game with perfect information. We thus restrict our attention to the class of pure-strategy MPEs.*

**Definition 3.2** (the value functions of the game). *Let  $(\pi_0^*, \pi_1^*)$  be a Markov perfect equilibrium, and for any state  $s \in \mathcal{S}$ , let  $V_0(s)$  and  $V_1(s)$  be the probabilities of the first-batting team winning and the last-batting team winning when in state  $s$ , respectively. That is,*

$$\begin{aligned} V_0(s) &= v_0(s; \pi_0^*, \pi_1^*), \quad s \in \mathcal{S}, \\ V_1(s) &= v_1(s; \pi_0^*, \pi_1^*), \quad s \in \mathcal{S}. \end{aligned}$$

*Then the functions  $V_0$  and  $V_1$  are called the value functions of the game for the first-batting team and the last-batting team, respectively.*

**Remark 3.2** (uniqueness of the value functions). *It is also well-known that the value function of the game for each team is unique. In other words, all equilibria must result in the same probability of the team winning.*

**Theorem 3.1** (Bellman equation). *The value functions satisfy the following recursive formula.*

$$V_0(s) = \begin{cases} r_0(s) & s \in \mathcal{S}_Q, \\ \text{Max}_{a \in \mathcal{A}(s)} \sum_{x \in \mathcal{X}(a)} V_0(t(s, a, x))p(x|s, a) & s \in \mathcal{S}_0, \\ \sum_{x \in \mathcal{X}(\pi_1^*(s))} V_0(t(s, \pi_1^*(s), x))p(x|s, \pi_1^*(s)) & s \in \mathcal{S}_1. \end{cases}$$

$$V_1(s) = \begin{cases} r_1(s) & s \in \mathcal{S}_Q, \\ \sum_{x \in \mathcal{X}(\pi_0^*(s))} V_1(t(s, \pi_0^*(s), x))p(x|s, \pi_0^*(s)) & s \in \mathcal{S}_0, \\ \text{Max}_{a \in \mathcal{A}(s)} \sum_{x \in \mathcal{X}(a)} V_1(t(s, a, x))p(x|s, a) & s \in \mathcal{S}_1. \end{cases}$$

$$\pi_0^*(s) \in \arg \max_{a \in \mathcal{A}(s)} \sum_{x \in \mathcal{X}(a)} V_0(t(s, a, x))p(x|s, a), \quad s \in \mathcal{S}_0. \quad (1)$$

$$\pi_1^*(s) \in \arg \max_{a \in \mathcal{A}(s)} \sum_{x \in \mathcal{X}(a)} V_1(t(s, a, x))p(x|s, a), \quad s \in \mathcal{S}_1. \quad (2)$$

**Theorem 3.2.** *The profile  $(\pi_0^*, \pi_1^*)$  of a Markov policy for the first-batting team satisfying (1), and a Markov policy for the last-batting team satisfying (2) is an MPE.*

## 4 Computational results

Table 1 shows the probability parameters for the starting order of the Fukuoka Softbank Hawks, and was compiled based on the values achieved in Japan’s professional baseball 2013 season [10]. Some of the stolen base success rates and sacrifice hit success rates for players with extremely small numbers of attempted stolen bases and sacrifice hits have been adjusted. Specifically, for the success rate of stolen bases for players with less than 4 attempted stolen bases over the year, the minimum of 0.7 and the actual value was adopted. Similarly, the minimum of 0.8 and the actual value was adopted for the sacrifice hit success rate for players with less than 4 attempted sacrifice hits over the year. The numbers in the Lineup column indicate the batting order for the players. Lineup A is the actual batting order which had been established by the closing stage of the 2013 season. Lineup B is a batting order created by shuffling them at random.

Table 1: Probability parameters

Name	Out	Single	Double	Triple	HR	Walk	Stolen Base success	Sacrifice Hit success	Lineup	
									A	B
Nakamura	0.608	0.208	0.0449	0.0041	0.0102	0.124	0.438	0.8*	1	3
Imamiya	0.719	0.154	0.0385	0.0067	0.0084	0.0737	0.769	0.925	2	5
Hasegawa	0.604	0.228	0.0459	0.0095	0.0301	0.0823	0.857	0	3	6
Uchikawa	0.621	0.202	0.0525	0.0016	0.0303	0.0924	0.7*	0	4	9
Matsuda	0.683	0.181	0.042	0.0081	0.0323	0.0533	0.65	0.8*	5	2
Yanagita	0.623	0.166	0.0564	0.0059	0.0326	0.116	0.909	0	6	7
Egawa	0.65	0.16	0.0152	0.0038	0.0456	0.125	0.5	0.6	7	4
Honda	0.661	0.207	0.0256	0.0085	0.0043	0.0938	0.786	0.864	8	8
Hosokawa	0.759	0.099	0.0438	0	0.0182	0.0803	0	0.833	9	1

A computational experiment was performed to find a MPE and the values of the games between lineup A and lineup B. The experimental results are as shown in Table 2.



Table 2: Values of the games

Bad first v.s. field fast	Pr(A wins)	Pr(B wins)	Pr(tie game)
Lineup A v.s. B	① 0.5008	② 0.4893	0.0099
Lineup B v.s. A	③ 0.5046	④ 0.4856	0.0098
Advantage of field fast	③−① 0.0038	②−④ 0.0037	—

- CPU Time: 1.26 sec / game
- Environment: a desktop PC with Intel® Core™ i7-3770K 16GB memory installed

And the winning percentages, under the conditions where there is a tie in the 8th inning and the both teams in the 9th inning start with batter number 1, are given in Table 3.

Table 3: Values of the games starting in the 9th inning

Bad first v.s. field fast	Pr(A wins)	Pr(B wins)	Pr(tie game)
Lineup A v.s. B	① 0.5079	② 0.4373	0.0548
Lineup B v.s. A	③ 0.5172	④ 0.4284	0.0544
Advantage of field fast	③−① 0.0093	②−④ 0.0089	—

In computational experiments, while performing a depth-first search of the “game tree” as an extensive form game of baseball, the recursive formula of Theorem 3.1 was solved for just those states that were reachable from the initial state  $s_0$ . Changing the run difference established in a called game from 30 runs to 60 runs and once again performing the computational experiments, the difference in the values of the game  $V_0(s_0)$  and  $V_1(s_0)$  were less than  $10^{-10}$ . It follows that we can reasonably consider the error from using the mercy-rule as practically zero.

We note that when the manager of each team makes decisions about hits, stolen bases, and sacrifice hits, the runs scored by the opposing team (the run difference) must be considered to win. For both lineup A and lineup B, if we subtract the winning percentage for each team when batting first from the winning percentage when batting last, then we get a positive value. This coincides with the result by Turocy [8]. Further, this difference appears to be more significant from the 9th inning onwards. This is because the last-batting team always has one more inning’s worth of information to study about the opposing team’s runs.

## 5 Summary and Future Challenges

In this paper, baseball has been formulated as a finite Markov game with approximately 3.5 million states. We demonstrated that the Markov perfect equilibria and the values of the games, where both teams’ managers maximize the probabilities of their respective team winning, can be computed in about one second per game.

In the future, we think that a raised level of analysis will be the object of the research, and not just the optimization of such strategies as sacrifice hits and stolen bases. For example, the authors are confident that the model given in this paper can be utilized to predict how the team’s winning percentage would change with the introduction and release of players, and in the sensitivity analysis of a player’s performance and the team’s winning percentage. At the same time as applying these using more detailed actual data, we would also like to investigate it’s use in other topics as challenges for the future.

## References

- [1] Bellman, R., Dynamic Programming and Markovian Decision Processes, with Application to Baseball in *Optimal Strategies in Sports*, S.P. Ladany and R.E. Macol(eds.), Elsevier-North Holland, New York, (1977), 77–85.
- [2] Bukiet, B., Harold, E.R. and Palacios, J. L., A Markov Chain Approach to Baseball, *Operations Research*, **45**(1), (1997), 14–23.
- [3] Cover, T. M. and Keilers, C. W., An Offensive Earned-Run Average for Baseball, , *Operations Research*, **25**(5), (1977), 729–740.
- [4] D’Esopo, D. A. and Lefkowitz, B., The Distribution of Runs in the Game of Baseball in *Optimal Strategies in Sports*, S.P. Ladany and R.E. Macol(eds.) , Elsevier North-Holland, (1977), 55–62.
- [5] Howard, R.A., *Dynamic Programming and Markov Processes* , M.I.T. Technology Press and Wiley, Cambridge, Mass, (1960).
- [6] Kira, A., Ueno, T., and Fujita, T., Threshold probability of non-terminal type in finite horizon Markov decision processes, *Journal of Mathematical Analysis and Applications*, vol. **386**, (2012), 461–472.
- [7] Shapley, L. S., Stochastic games, *Proceedings of the National Academy of Sciences of the United States of America*, **39**, (1953), 1095-1100.
- [8] Turocy, T. L., In Search of the “Last-Ups” Advantage in Baseball: A Game-Theoretic Approach, *Journal of Quantitative Analysis in Sports*, **4**(2) , (2008), Article 5.
- [9] Zachrisson, L. E., Markov games. *Annals Math. Studies*, **52**. *Advances in Game Theory*, M. Drescher, L.S. Shapley and A.W. Tucker(eds.), Princeton University Press, Princeton, 1964.
- [10] Let’s enjoy baseball data! (in Japanese), (<http://baseballdata.jp/>), Accessed 2014 Mar 25.