VU VRIJE UNIVERSITEIT AMSTERDAM

VU Research Portal

A real-world stochastic two-person game

Tijms, H.C.; van der Wal, J.T.

published in
Probability in the Engineering and Informational Sciences
2006

DOI (link to publisher)
10.1017/S0269964806060372

document version
Publisher's PDF, also known as Version of record

Link to publication in VU Research Portal

citation for published version (APA)
Tijms, H. C., & van der Wal, J. T. (2006). A real-world stochastic two-person game. *Probability in the Engineering and Informational Sciences*, *20*, 599-608. https://doi.org/10.1017/S0269964806060372

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 02. Dec. 2021

Probability in the Engineering and Informational Sciences, **20**, 2006, 599–608. Printed in the U.S.A.

A REAL-WORLD STOCHASTIC TWO-PERSON GAME

HENK TIJMS
Department of Econometrics and Operations Research
Vrije University
Amsterdam, The Netherlands
E-mail: tijms@feweb.vu.nl

JAN VAN DER WAL
Department of Quantitative Economics
Faculty of Economics and Econometrics
University of Amsterdam
Amsterdam, The Netherlands
and
Department of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
E-mail: jan.v.d.wal@tue.nl

This article discusses a real-world application of a terminating two-person stochastic game. The problem comes from a Dutch television game show in which two finalists play a dice game. Each player chooses a number of dice to be rolled. The score of the roll is added to the player's total provided that none of the dice showed the outcome one. The first player reaching a prespecified number of points is the winner. This article discusses the computation and the structure of an optimal strategy.

1. INTRODUCTION

This article deals with a real-world problem that might seem of recreational nature at first sight but that offers challenging questions of a general nature in a terminating two-person stochastic game. The problem concerns a real-world situation arising during a television game show. At the end of the show, the two remaining

© 2006 Cambridge University Press 0269-9648/06 $16.00

contestants have to play a two-person dice game. The contestants each sit behind a panel with a battery of buttons numbered $1,2,\ldots,D$ (say, $D = 10$). In each stage of the game, all of the contestants must simultaneously press one of the buttons and the contestants cannot observe each other's decision. The number on the button pressed by the contestant is the number of dice that are thrown for the contestant. For each contestant, the score of the throw for that contestant is added to his/her total, provided that none of the dice in that throw showed the outcome 1; if the outcome is 1, no points are added to the current total of the candidate. The candidate who first reaches a total of G (say, $G = 100$) points is the winner. In the case that both candidates reach the goal of G points in the same move, the winner is the candidate who has the largest total. In the event of a tie, the winner is determined by a toss of a fair coin. At each stage of the game, both candidates have full information about his/her own current total and the current total of the opponent. The formulation of the game is such that it is zero-sum and stochastic. What does the optimal strategy look like? Do random actions appear? If so, when?

2. SOME PRELIMINARIES

Let us first look at the distribution of the number of points earned in a single throw with d dice. Define the random variable Y_d as

$$Y_d := \text{the number of points added to a contestant's total, throwing } d \text{ dice.}$$

Letting the random variable X_i denote the outcome shown by the ith dice, Y_d equals $X_1 + \cdots + X_d$ if none of $X_1,\ldots,X_d$ equals 1, and Y_d is 0 otherwise. The random variables $X_1,\ldots,X_d$ are independent and identically distributed (i.i.d.). Moreover, given that X_i is not 1, the conditional distribution of X_i is the uniform distribution on the integers $2,3,\ldots,6$. This conditional distribution has expected value 4. The probability of not getting a 1 in a throw of d dice is $(\frac{5}{6})^d$. Elementary calculations show that

$$E(Y_d) = \left(\frac{5}{6}\right)^d 4d \quad \text{and} \quad \text{var}(Y_d) = \left(\frac{5}{6}\right)^d (16d^2 + 2d) - \left(\frac{5}{6}\right)^{2d} (4d)^2.$$

The maximum of $E(Y_d)$ is easily found by looking at

$$E(Y_{d+1}) - E(Y_d) = \left(\frac{5}{6}\right)^d 4\left(\frac{5}{6}(d+1) - d\right).$$

The difference is positive for $d < 5$, is zero for $d = 5$, and is negative for $d > 5$. Hence, $E(Y_d)$ is maximized by taking d equal to 5 or 6. Heuristically, this result can also be obtained by marginal analysis. Given that you already have d dice in your hand, should you pick up another one? If one of the previous dice will give a 1, it is irrelevant what you do, so assume none of the other dice will give a 1. Then, on average, every one of them will contribute four points. So, in this situation with probability 1/6, you lose $4d$ points, and with probability 5/6, you win another four

points. Thus, the expected gain is $20/6 - 4d/6$ points, which becomes zero for $d = 5$.

Table 1 gives the probability $P(Y_d > 0)$ together with the mean μ_d and standard deviation σ_d of the random variable Y_d for various values of d.

If you consider only the mean, the optimal number of dice is five or six for the situation of a single move. However, as the standard deviation shows, throwing with five or six dice is not the same. With six dice, the throw will be "riskier." If you quickly need many points, then you have to take a risk so that throws with seven or more dice come into the picture.

Next, we discuss how to compute the probability distribution of Y_d. For any $d \geq 1$, let

$$q_i^{(d)} = P(Y_d = i) \quad \text{and} \quad r_i^{(d)} = P(Y_d = i | Y_d > 0) \quad \text{for } i = 0,1,\ldots.$$

Obviously,

$$q_0^{(d)} = 1 - \left(\frac{5}{6}\right)^d \quad \text{and} \quad q_i^{(d)} = \left(\frac{5}{6}\right)^d r_i^{(d)} \quad \text{for } i,d = 1,2,\ldots$$

and

$$r_i^{(d)} = \sum_{j=2}^{6} \frac{1}{5} r_{i-j}^{(d-1)}, \qquad i = 2d, 2d+1, \ldots, 6d, \quad \text{and} \quad r_i^{(d)} = 0 \text{ otherwise,}$$

with the convention $r_0^{(0)} = 1$ and $r_i^{(0)} = 0$ for $i \neq 0$.

3. TWO ONE-PERSON GAMES

To get some insight, let us consider the following two one-person games. In the first one, we minimize the expected number of throws needed to reach G points. In the second one, we maximize the probability of reaching G in a given number of throws.

TABLE 1. Mean μ_d, Standard Deviation σ_d, and the Probability $P(Y_d > 0)$

d	μ_d	σ_d	$P(Y_d > 0)$	d	μ_d	σ_d	$P(Y_d > 0)$
1	3.3333	1.9720	0.8333	7	7.8143	12.7139	0.2791
2	5.5556	4.0445	0.6944	8	7.4422	13.6559	0.2326
3	6.9444	6.2113	0.5787	9	6.9770	14.3521	0.1938
4	7.7160	8.2327	0.4823	10	6.4602	14.8292	0.1615
5	8.0376	10.0084	0.4019	20	2.0867	12.7917	0.0261
6	8.0376	11.5029	0.3349	30	0.5055	7.7885	0.0042

3.1. Expected Number of Throws

Define $V(i)$ as the minimal expected number of additional throws when one still needs i additional points to reach the goal. Then, letting $V(i) = 0$ for $i \leq 0$, we have the dynamic programming equation (cf. Derman [1])

$$V(i) = \min_d \left\{ 1 + q_0^{(d)} V(i) + \sum_{j=2d}^{6d} q_j^{(d)} V(i-j) \right\}$$

or, equivalently,

$$V(i) = \min_d \left\{ \frac{1}{1 - q_0^{(d)}} \left[1 + \sum_{j=2d}^{6d} q_j^{(d)} V(i-j) \right] \right\}.$$

Table 2 gives the minimal expected number of throws and $d^*(i)$, the optimal number of dice to use in state i, where $D = 10$ is the maximum number of dice that can be thrown.

As we see, the number of dice to use varies quite a lot. Even using seven dice is optimal in some states. Also note that in each of the states $i = 1, \ldots, 28$ in which the goal is to get at least i additional points, you use the number of dice that will hopefully give you i or more points, when there is no one among the outcome of the throw with those dice. In other words, in each state i with $1 \leq i \leq 28$, you try to reach the goal in one "successful" throw. In each of the states i with $29 \leq i \leq 40$, however, the optimal strategy is to reach the goal in two "successful" throws instead of one. Of course, if i gets large, you will only use throws with five or six dice, as these throws have the highest expected "value" (recall Table 1). As a result of this, we must also have $\lim_{i\to\infty} V(i+1) - V(i) = 1/8.0376$, where 8.0376 is the expected number of points per throw with five or six dice.

Table 2. Minimizing the Expected Number of Throws for $D = 10$

i	$V(i)$	$d^*(i)$	i	$V(i)$	$d^*(i)$	i	$V(i)$	$d^*(i)$	i	$V(i)$	$d^*(i)$
1	1.2000	1	11	2.0836	3	21	3.1929	6	31	4.5383	4
2	1.2000	1	12	2.1427	4	22	3.3010	6	32	4.6426	5
3	1.4400	2	13	2.2138	4	23	3.4355	6	33	4.7438	5
4	1.4400	2	14	2.3218	4	24	3.5919	6	34	4.8484	5
5	1.4880	2	15	2.4674	4	25	3.7622	6	35	4.9568	5
6	1.5840	2	16	2.5876	5	26	3.9239	7	36	5.0690	5
7	1.7376	3	17	2.6644	5	27	4.0539	7	37	5.1850	5
8	1.7664	3	18	2.7729	5	28	4.2027	7	38	5.3049	5
9	1.8259	3	19	2.9129	5	29	4.3254	4	39	5.4287	5
10	1.9277	3	20	3.0787	5	30	4.4300	4	40	5.5565	5

3.2. Limited Number of Throws

Define $p^{(l)}(i)$ as the maximal probability of getting at least i additional points in at most L throws. Then, using standard dynamic programming, we have

$$p^{(l)}(i) = \max_d \left\{ \sum_j q_j^{(d)} p^{(l-1)}(i-j) \right\},$$

where $p^{(l)}(i) = 1$ for $i \leq 0$ and $p^{(0)}(i) = 0$ for $i > 0$.

The results of a maximization with $D = 10$ are given in Table 3 for several values of the limit l on the number of allowed throws. In Table 3, $d^*(i, l)$ is the optimal number of dice to throw when you still need i points and you have l throws left. As we see, the number of dice to use is more regular (i.e., varies in a more monotonic way than in the case of minimizing the expected number of throws). You also see that starting at 40 with six throws left, you throw four dice. If the score turns out to be zero, you continue with five dice in the next throw. If then the score is 17 (so you have reached 23), you continue with four dice again, but if your score is 22 (so you are in 18), you continue with three dice, and so on.

It can be noted from Table 3 that for fixed values of i the maximal probability $p^{(l)}(i)$ as function of l has an S-shaped form; that is, for each i, there is an $l(i)$ such that $p^{(l)}(i)$ is convex for $l < l(i)$ and concave for $l > l(i)$. Computations of $p^{(l)}(i)$ for larger values of i and l affirmed this behavior. It is our conjecture that, indeed, $p^{(l)}(i)$ is always S-shaped.

To the contrary, the behavior of $p^{(l)}(i)$ for fixed l as a function of i is quite irregular. We have also found that the somewhat surprising result that for l equal to 5 or 6, the value of $d^*(i, l)$ is not monotone. For $l = 5$, it first increases to 10 (the maximal number of dice) for $i = 79, \ldots, 89$, then decreases to 9 for $i = 90, 91$, then to 8 for $i = 92, 93, 94$, after which it again increases via 9 to 10. For $l = 6$, it increases to 8 for $i = 74, \ldots, 82$, then decreases to 7 for $i = 83, \ldots, 91$, after which it again increases via 8 and 9 to 10.

4. THE TWO-PERSON STOCHASTIC GAME

The rules of the game state that in each throw, simultaneously the two players have to decide on the number of dice to use, without seeing what the opponent is doing but knowing and using the scores so far. So, after a number of throws, player 1 still needs a points and player 2 needs b points. Thus, the state space is two dimensional. If now player 1 decides to use k dice and player 2 uses l, then the state changes from (a, b) to $(a - i, b - j)$ with probability $q_i^{(k)} q_j^{(l)}$.

The game is a stochastic terminating zero-sum game. If we assume that the number of dice to be used in each throw is limited by some number, D say ($D = 10$ in the television game show), then the game can be solved recursively by dynamic programming.

TABLE 3. Probability of Getting at Least i Points in at Most L Throws ($D = 10$)

i	$p^{(1)}(i)$	$d^*(i,1)$	$p^{(2)}(i)$	$d^*(i,2)$	$p^{(3)}(i)$	$d^*(i,3)$	$p^{(4)}(i)$	$d^*(i,4)$	$p^{(5)}(i)$	$d^*(i,5)$	$p^{(6)}(i)$	$d^*(i,6)$
1	0.833	1	0.972	1	0.995	1	0.999	1	1.000	1	1.000	1
2	0.833	1	0.972	1	0.995	1	0.999	1	1.000	1	1.000	1
3	0.694	2	0.921	1	0.982	1	0.996	1	0.999	1	1.000	1
4	0.694	2	0.907	2	0.975	1	0.994	1	0.999	1	1.000	1
5	0.667	2	0.894	2	0.967	2	0.990	1	0.997	1	0.999	1
6	0.611	2	0.867	2	0.957	2	0.987	2	0.996	1	0.999	1
7	0.574	3	0.838	2	0.945	2	0.982	2	0.994	2	0.998	1
8	0.560	3	0.812	3	0.930	2	0.976	2	0.992	2	0.998	2
9	0.532	3	0.795	3	0.916	2	0.969	2	0.990	2	0.997	2
10	0.486	3	0.765	3	0.898	3	0.960	2	0.986	2	0.995	2
11	0.471	4	0.743	3	0.885	3	0.952	2	0.982	2	0.994	2
12	0.455	4	0.716	3	0.868	3	0.941	3	0.977	2	0.992	2
13	0.428	4	0.693	4	0.851	3	0.932	3	0.972	2	0.989	2
14	0.395	5	0.667	4	0.830	3	0.921	3	0.965	2	0.986	2
15	0.386	5	0.649	4	0.812	3	0.909	3	0.959	3	0.983	2
16	0.370	5	0.625	4	0.793	4	0.897	3	0.952	3	0.979	2
17	0.346	5	0.597	5	0.773	4	0.883	3	0.944	3	0.975	2
18	0.325	6	0.577	5	0.755	4	0.869	3	0.936	3	0.970	3
19	0.316	6	0.561	5	0.738	4	0.855	3	0.927	3	0.965	3
20	0.302	6	0.539	5	0.716	4	0.839	4	0.917	3	0.959	3
21	0.281	6	0.510	5	0.691	4	0.822	4	0.905	3	0.953	3
22	0.268	7	0.494	6	0.675	5	0.808	4	0.895	3	0.946	3
23	0.259	7	0.480	6	0.659	5	0.794	4	0.884	3	0.939	3
24	0.246	7	0.460	6	0.639	5	0.778	4	0.872	3	0.931	3
25	0.229	7	0.436	6	0.616	5	0.759	4	0.858	4	0.923	3
26	0.220	8	0.421	7	0.598	5	0.742	4	0.846	4	0.914	3
27	0.213	8	0.408	7	0.581	5	0.726	4	0.833	4	0.905	3
28	0.202	8	0.391	7	0.563	6	0.708	5	0.820	4	0.895	3
29	0.187	8	0.371	7	0.542	6	0.690	5	0.805	4	0.884	3
30	0.182	9	0.356	8	0.525	6	0.674	5	0.792	4	0.874	4
31	0.175	9	0.345	8	0.510	6	0.659	5	0.778	4	0.864	4
32	0.165	9	0.331	8	0.493	6	0.642	5	0.763	4	0.853	4
33	0.154	10	0.314	8	0.473	6	0.623	5	0.747	4	0.841	4
34	0.150	10	0.301	8	0.457	7	0.607	5	0.732	4	0.829	4
35	0.144	10	0.291	9	0.443	7	0.591	5	0.717	5	0.817	4
36	0.136	10	0.278	9	0.428	7	0.574	5	0.702	5	0.804	4
37	0.126	10	0.264	9	0.411	7	0.557	6	0.687	5	0.791	4
38	0.115	10	0.248	9	0.392	7	0.538	6	0.670	5	0.778	4
39	0.102	10	0.230	10	0.373	7	0.520	6	0.653	5	0.763	4
40	0.088	10	0.214	10	0.355	7	0.501	6	0.636	5	0.749	4

The value of the game is equal to the probability that player 1 wins minus the probability that player 2 wins, given that both players play optimally. Define

$$V(a,b) = \begin{cases} 1 & \text{if } a < b \text{ and } a \le 0 \\ 0 & \text{if } a = b \le 0 \\ -1 & \text{if } a > b \text{ and } b \le 0. \end{cases} \quad \textbf{(1)}$$

We want to determine $V(a,b)$ for both a and b positive (and less than or equal to G) and the optimal, possibly randomized, actions that guarantee this value.

4.1. Randomized Actions

The first question might be: Do the players have to randomize the number of dice to use in a throw? Some insight is already gained by just looking at the game starting in (1,1). Knowing that the value of this symmetric state has to be zero, we can check whether there is a "deterministic move" (i.e., using a fixed number of dice) that guarantees the value zero. If there would be an optimal deterministic throw, then we should have for some $d \le D$ and for all l,

$$V^{(d,l)}(1,1) := \frac{1}{1 - q_0^{(d)} q_0^{(l)}} \sum_{i,j;\, i+j>0} q_i^{(d)} q_j^{(l)} V(1-i, 1-j) \ge 0.$$

Computing $\min_l V^{(d,l)}(1,1)$ for all d leads to the results in Table 4.

So, there is no optimal number of dice. The best number is four, but even then, the best you can get is -0.0649. If your opponent knows the number of dice you use, it is optimal for him to use one die more, unless you use five or more dice; then his optimal choice is one die. Thus, randomization is necessary.

4.2. The Optimality Equation

The two-person zero-sum stochastic game is in fact a terminating, even contracting game. In each move (throw of the two players), the state of the game gets closer to the payoff zone: the set of states (a,b) with $\min\{a,b\} \le 0$ (define the distance from (a,b) to the payoff zone as $a+b$ if both a and b are positive; then with a probability of at least $1 - (q_0^{(D)})^2$, the distance decreases by at least 2).

The value of the game and the optimal moves of the two players can be computed by repeatedly solving the appropriate matrix games. Let $x = (x_1, x_2, \ldots, x_D)$ be a randomized move for player 1; that is, player 1 throws d dice with probability

TABLE 4. Best Result for Player 1 Restricting to Deterministic Moves

d	$\min_l V^{(d,l)}(1,1)$	Best Response to d
1	−0.3951	2
2	−0.2282	3
3	−0.1282	4
4	−0.0649	5
5	−0.1072	1
6	−0.2467	1
7	−0.3656	1
8	−0.4666	1
9	−0.5522	1
10	−0.6245	1

x_d, where $\sum_d x_d = 1$. The first approach to think of is to recursively compute $V(a,b)$ via a sequence of LP-problems, starting in $(a,b) = (1,1)$ and working backward, step by step, until $(a,b) = (G,G)$. This requires solving the optimization problem:

$$\text{maximize } V \qquad \text{subject to}$$

$$\sum_d x_d \left(\sum_{i+j>0} q_i^{(d)} q_j^{(l)} V(a-i, b-j) + q_0^{(d)} q_0^{(l)} V \right) \geq V, \qquad l = 1, \ldots, D,$$

$$x_d \geq 0, \quad d = 1, \ldots, D, \quad \sum_d x_d = 1,$$

where, for $i + j > 0$, the values $V(a-i, b-j)$ have been computed earlier and, hence, are known. (V is unrestricted in sign.) However, this optimization problem is not exactly an LP-problem because of the nonlinear term $\sum_d x_d q_0^{(d)} q_0^{(l)} V$.

To make an LP-approach possible, we proceed as follows. Define $V^{(n)}(a,b)$ as the value of the game if it is played at most n times with a terminal reward 0, if after n steps the game has not yet reached the payoff zone. Thus, $V^{(0)}(a,b) := 0$ if $a > 0$ and $b > 0$. Also, define

$$V^{(n)}(a,x,b,l) = \sum_d x_d \sum_{i,j} q_i^{(d)} q_j^{(l)} V^{(n-1)}(a-i, b-j), \qquad n > 0,$$

with the convention that, for $n \geq 0$ and $a \leq 0$ or $b \leq 0$, $V^{(n)}(a,b) = V(a,b)$ with $V(a,b)$ as defined in (1). Then, in iteration n for state (a,b), the value of the game and an optimal move for player 1 can be obtained from the following LP-problem for a matrix game (cf. Maitra and Sudderth [2]):

$$\text{maximize } V \qquad \text{subject to}$$

$$V^{(n)}(a,x,b,l) \geq V, \qquad l = 1, \ldots, D,$$

$$x_d \geq 0, \quad d = 1, \ldots, D, \quad \sum_d x_d = 1.$$

The optimal value V satisfies $V = V^{(n)}(a,b)$ and the optimal $x^{(n)}(a,b)$ represents an optimal move for player 1 in state (a,b) in iteration n. $V^{(n)}(a,x,b,l)$ converges exponentially fast to the value of the game, and $x^{(n)}$ is nearly optimal for n sufficiently large. Similarly, we can compute a (nearly) optimal strategy for player 2. Of course, for symmetry reasons, the optimal move for player 2 in (a,b) is the same as the optimal move for player 1 in (b,a).

Remark. In order to profit from the contracting properties of the dynamic programming scheme for V^n, one could introduce a so-called weighted supremum norm μ. Defining $\mu(a,b) = \alpha^{a+b}$ for some $\alpha > 1$, the model will be contracting with respect to the μ-norm and nearly optimal strategies and upper and lower bounds can be

TABLE 5. Optimal Strategy for Player 1 in (k, l) with $1 \leq k, l \leq 13$

	13	12	11	10	9	8	7	6	5	4	3	2	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0.144	0	0	0	0.238	0	0
	1	1	1	1	1	1	0.856	1	1	1	0.762	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0.366	0.140	0	0	0	0.233	0
	1	1	1	1	1	1	0.634	0.860	1	1	1	0.767	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.203	0.089	0	0	0.268	0.156
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0.797	0.911	1	1	0.732	0.844
9	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.277	0.196	0.085	0	0	0.259
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0.723	0.804	0.915	1	1	0.741
8	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.335	0.272	0.192	0.083	0	0.330
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0.665	0.728	0.808	0.917	1	0.670
7	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0.332	0.269	0.190	0.082	0
	1	0.091	0.006	0	0	0	0	0	0	0	0	0	0
	0	0.909	0.994	1	1	1	1	1	0.668	0.731	0.810	0.918	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0.226	0.209	0.198	0.172	0.076
	0	0	0	0	0	0	0	0	0	0.073	0.042	0	0
	1	1	0.178	0.101	0.093	0.008	0	0	0	0	0	0	0
	0	0	0.822	0.899	0.907	0.992	1	1	0.774	0.718	0.760	0.828	0.924
5	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0.202	0.196	0.187	0.166
	0	0	0	0	0	0	0	0.016	0	0.058	0.081	0.046	0
	1	1	1	0.260	0.260	0.183	0.054	0	0	0	0	0	0
	0	0	0	0.740	0.740	0.817	0.946	0.984	1	0.740	0.723	0.767	0.834
4	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.064	0.127	0.176	0.196	0.190	0.182
	0	0	0	0	0	0	0	0	0.043	0.053	0.064	0.084	0.048
	1	1	1	1	0.445	0.358	0.229	0.100	0	0	0	0	0
	0	0	0	0	0.555	0.642	0.771	0.836	0.830	0.771	0.740	0.726	0.770
3	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.064	0.137	0.146	0.176	0.196	0.190
	1	0.001	0	0	0	0	0	0	0	0.075	0.053	0.065	0.084
	0	0.999	1	1	1	0.556	0.408	0.227	0.134	0	0	0	0
	0	0	0	0	0	0.444	0.592	0.709	0.729	0.779	0.771	0.739	0.726
2	0	0	0	0	0	0	0	0	0	0	0	0	0.073
	0	0	0	0	0	0	0	0	0.142	0.172	0.146	0.176	0.067
	1	1	0.070	0.015	0	0	0	0	0	0	0.075	0.053	0.172
	0	0	0.930	0.985	1	1	0.612	0.577	0.204	0.151	0	0	0
	0	0	0	0	0	0	0.388	0.423	0.654	0.677	0.779	0.771	0.688
1	0	0	0	0	0	0	0	0	0	0	0	0.030	0
	0	0	0	0	0	0	0	0	0	0.180	0.172	0.171	0.176
	1	1	1	0.162	0.143	0.059	0	0	0	0	0	0	0.053
	0	0	0	0.838	0.857	0.941	1	0.817	0.760	0.194	0.151	0.168	0
	0	0	0	0	0	0	0	0.183	0.240	0.626	0.677	0.631	0.771

computed from the difference between $V^{(n+1)}$ and $V^{(n)}$ (cf. van der Wal and Wessels [3]).

4.3. Numerical Results

In Table 5 we present some results for the optimal strategy for the case that the maximum number of dice that can be thrown is $D = 5$. Table 5 should be read as follows. If, for instance, player 1 still needs one point and player 2 needs three points, then player 1 will use two, four, or five dice with probabilities 0.172, 0.151, and 0.677, respectively.

Our numerical results indicate that the optimal strategy has certain monotonicity properties. It would be nice to have a theoretical proof for the monotonicity properties. Also, it appears from the calculations in Table 5 that for states above 13, the players use nonrandomized decisions only. Again, this is a kind of turnpike result one would expect, but it would be nice if it could be proved theoretically. What we also see from Table 5 is the following. For instance, in state (5,13), player 1 will use four dice and player 2 will use five dice. So both players use more dice than needed to reach the payoff zone in order to beat the other player in case neither one of them throws a 1.

5. VARIANTS

There are various possible modifications of this game:

1. Player 2 uses the optimal strategy with respect to one of the one-person games discussed earlier. What is the optimal response for player 1 and how does this increase "his value"? This "game" can still be solved by ordinary dynamic programming.
2. Suppose that the players alternatingly throw a number of dice, where at the beginning of the show, a coin is flipped to decide which player starts. Again, a simple dynamic programming algorithm suffices to obtain the optimal strategy.
3. Suppose that a player gets not only a score of zero but also loses all (or some of) the points collected so far if there is an outcome 1 in the throw of his dice.
4. Suppose the players know the outcomes of their own throws, but do not know what the other player has been doing at all. This is a game with imperfect information. Is it possible to determine an optimal strategy?
5. Suppose that, in addition to the previous situation, you also know how many dice your opponent has used. This, too, is a game with imperfect information.

References

1. Derman, C. (1970). *Finite state Markovian decision problems*. New York: Academic Press.
2. Maitra, A. & Sudderth, D. (1996). *Discrete gambling and stochastic games*. Berlin: Springer-Verlag.
3. Van der Wal, J. & Wessels, J. (1977). Successive approximation methods for Markov games. In H.C. Tijms and J. Wessels (eds.), *Markov decision theory*, pp. 39–55. Amsterdam: CWI.