Special Proceedings of the 22nd Annual Conference of SSCA held at S. P. Pune University, during January 02–04, 2020; pp 103–116

Optimal Decisions Under Partial Refunds: A Reward-Earning Random Walk on a Parity Dial

Jyotirmoy Sarkar
Department of Mathematical Sciences
Indiana University–Purdue University Indianapolis, Indiana, USA

Received: 20 July 2020; Revised: 31 July 2020; Accepted: 03 August 2020

Abstract

We solve an unsolved problem posed in Sarkar (2020), which proposed a reward-earning binary random walk game on a parity dial whose twelve nodes, when read clockwise, are labeled as $(1, 11, 3, 9, 5, 7, 6, 8, 4, 10, 2, 0)$. Starting from Node 0, at each step the player tosses a fair coin and moves one step clockwise (if heads) or counterclockwise (if tails). The player pays $25c + k$ cents if she intends to capture c nodes and toss the coin k times. When the c non-zero nodes are captured or when the k tosses are over the game ends; and the player earns as many nickels as the sum of the labels of the captured nodes. The player's objective is to determine (c, k) to minimize the expected percentage loss.

Here we consider a more complex game in which the player is offered several options for a partial refund on each unused toss on payment of an additional upfront overhead fee. Which partial refund offer should she choose? Having chosen the refund option, how should she determine (c, k) to minimize the expected percentage loss?

Under partial refund offers, the player may choose a higher c and a higher k compared to those in the no refund scenario. The optimal choice is discovered through computer simulation, leaving open the theoretical development. Lessons learned from such games empower all parties engaged in the marketplace to determine when to intervene and how to make decisions to benefit from an opportunity and/or prevent a catastrophe.

Key words: Bernoulli variable; Reward random walk; Stopping time; Guaranteed refund.

AMS Subject Classifications: 60G50, 05C81

1. Introduction

It ought to be a common knowledge that when a casino offers you a game of chance and you agree to play, *on average* you should expect to lose money: For otherwise, the Casino would simply toss the game out. You willingly accept this anticipated loss in exchange for deriving some entertainment pleasure and experiencing the excitement of winning a big windfall (although that would happen only rarely). The casino must make money even after paying windfalls, costs, staff salaries, subsidies and taxes. The lure of a game is irresistible

Corresponding Author: Jyotirmoy Sarkar
Email: jsarkar@iupui.edu

when the game *appears to be* in favor of the player for then the casino can entice more players play it more often, and earn more profit for itself. The casino, of course, knows the exact long run prospects of each game it offers. Sarkar (2020) proposed and analyzed such a game, but left as unsolved a more realistic, and more complex, problem of how to choose among several refund policies. Here we take up that generalized problem and discover the optimal choice for the player.

Both the original game and the generalized game serve as models for entrepreneurial decisions and consumer choices. In repeated plays of the game, the optimal choice for each party may be discovered by utilizing the theory of stochastic processes. We direct interested readers to Ross (1996) and Medhi (1982) to encounter the general theory of stochastic processes, to Lovasz (1993) to learn about random walks on graphs, and to Maiti and Sarkar (2019) to study symmetric random walks on paths and cycles. However, to communicate better with researchers outside mathematical sciences, here we rely on computer simulations to discover the optimal choices. Lessons learned from the game will empower all parties engaged in the marketplace to determine when and how to intervene in order to maximally benefit from an opportunity and/or prevent a catastrophe.

In Section 2, we describe the original game proposed in Sarkar (2020) and summarize the optimal choice for the player. In Section 3, we describe the modified game and an expedited search algorithm to conduct the simulation study. In Section 4, we study each refund option and discover the optimal choice of (c, k) through simulation. Section 5 gives the properties of the optimal game within the optimal refund option. Section 6 translates the lessons learnt from this generalized game of reward-earning binary random walk to decision making in the marketplace.

All computations are done using the freeware R, and codes are given in the Annexure.

2. The Original Game

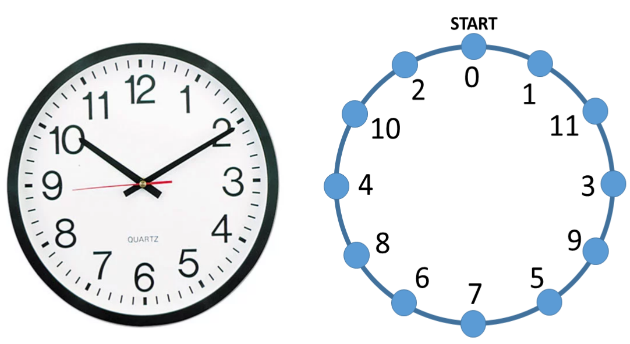

Figure 1: The usual dial of a clock and the parity dial

On a circle there are twelve nodes labeled $(1, 11, 3, 9, 5, 7, 6, 8, 4, 10, 2, 0)$ going clockwise, as shown in Figure 1. The labels are obtained from the usual dial of a clock by changing the top node from 12 to 0 and by interchanging nodes within pairs $(2, 11)$, $(4, 9)$, and $(6, 7)$. Thus, all odd values are on the right half while all even values are on the left half of the dial. Hence, this dial is called the parity dial.

To understand the nature and properties of a binary random walk on any dial, we refer the interested reader to Sarkar (2006). If this random walk also produces earned rewards when a specific node is visited it becomes a reward-earning random walk. Such a reward-earning binary random walk on the parity dial studied in Sarkar (2020): A player pays an admission price of $25c + k$ cents, where c is the number of nodes she intends to visit and capture and k is the number of times she wishes to toss a fair coin. The player begins at Node 0. After each toss, the player moves one step clockwise if the outcome is heads (with probability half), or one step counterclockwise if the outcome is tails; and she captures a node on the first visit to it. The game ends as soon as either c nodes (other than Node 0) are captured or k tosses are over. Then the player will earn as many nickels as the sum of the labels of the captured nodes. How should the player determine c and k?

The choice of $c = 1$ is immediately ruled out because then the player must choose $k = 1$, and pay 26 cents per play. She will earn either one or two nickels with probability half each, or on average, $(5 + 10)/2 = 7.5$ cents. Therefore, she will lose 18.5 cents—a whopping 71.2% loss! Next, for $c = 2$, Sarkar (2020) proves that the optimal choice of k is 6; and in that case, the player stands to lose about 11.7 cents or 21% of her wager of 56 cents. Thereafter, for $3 \leq c \leq 11$, he obtains the optimal k via simulation (based on 10^5 iterations). We summarize his results in Table 1.

Table 1: For the game with no refund, the optimal k's for each $3 \leq c \leq 11$, determine the optimal choice of (c, k) as $(6, 28)$ with an expected loss of 9.69% (marked by a †).

			cents		
c	k	price	E[rew]	E[loss]	E[%loss]
3	10	85	70.13	14.87	17.50
4	16	116	102.27	13.73	11.84
5	22	147	131.53	15.47	10.53
† 6	28	178	160.76	17.24	† 9.69
7	36	211	189.72	21.28	10.09
8	44	244	220.29	23.71	9.72
9	54	279	248.65	30.35	10.88
10	64	314	279.96	34.04	10.84
11	72	347	304.72	42.28	12.18

Based on Table 1, we learn that the gambler's best choice game is $(c = 6, k = 28)$; and with this choice, she faces a 9.69% expected loss. A gambler with a tolerance limit of 10% loss can play this game. The only other choice within her tolerance limit is $(c = 8, k = 44)$ with a 9.72% loss. It is somewhat perplexing that $(c = 7, k = 36)$ results in a higher expected loss of 10.09% than either $(c = 6, k = 28)$ or $(c = 8, k = 44)$. But this can be explained by noting that the node labels an odd distance away from Node 0 are typically smaller than node labels at an even distance.

3. The Generalized Game with Refund Options

To understand the need and the nature of the generalized game, let me paraphrase a very productive conversation I had with one attentive listener when I gave a talk on the above-stated original game at the 2020 Pune Conference of the SSCA. During high tea after the conference ended, among other things, this compassionate (to a gambler who would play this game) and extravagantly helpful (to me) gentleman spoke to me thus:

> "When the gambler captures c nodes, she is happy. But wouldn't she feel poorly about forfeiting all the unused tosses she already has paid for?"
>
> "Indeed, she would. That is the very essence of the decision-making problem in choosing both c and k optimally. Having chosen a c, if the gambler picks too small a k, she will likely not have captured all c nodes when she has tossed k times. On the other hand, if she picks too large a k, by the time she has captured c nodes she would have many unused tosses which she would forfeit. She must choose k cautiously."
>
> "You mentioned offering a guaranteed refund in exchange for the unused tosses. If you refund a full penny for every unused toss, the game surely becomes more attractive to the player. It may even become favorable to the gambler! Is that something the Gambling House will allow?"
>
> "A guaranteed refund does not mean a full refund. If it did, then the gambler would simply pay for $k = 1000$ (or a large number thereabouts) knowing that there is no risk of losing the excess payment. She would recover it all as soon as she captures c nodes, which will happen with almost certainty. For all practical purposes, one can think as if the player pays $25c$ at the start of the game and then pays one penny before each toss until c nodes are captured, requiring a random number of tosses K_c. Alternatively, the payment of $25c + K_c$ can be determined when the game ends. In either case, the problem changes to choosing c alone. Furthermore, the Gambling House will likely charge the player an upfront fee to purchase this option to get a 100% refund."

I had not calculated the optimal c under the full refund scenario with or without any fee since a 100% refund option was not on my mind prior to this conversation. Therefore, I could not talk about the optimal choice of c, except to say that it is likely to be an even number 6 or more, and to reiterate that that is not what I meant by a guaranteed refund.

For the benefit of my readers, I have since then carried out that missing simulation. R codes are given in the Annexure. Table 2 summarizes the expected percentage loss for various choices of c under 100% refund at overhead fees $0, 5, 10, \ldots, 30$ cents. I should point out that in this scenario, the (random) number of tosses is not right truncated by a predetermined k as in the original game; hence, the price paid is genuinely random (and it is determined when the game ends with c nodes captured). Hence, the expected percentage loss is calculated only after the game ends using the formula

$$\text{E}[\%\text{loss}] = 100 \times \frac{\text{E[loss] + overhead fee}}{\text{E[price] + overhead fee}}$$

Bear in mind that the very definition of expected loss has changed! One could pretend as if 100 extra tosses are paid for and the extra payment is recovered as refund, in which

Table 2: Determining optimal c (marked by a *) under a 100% refund option bought upfront on payment of some overhead fee

	cents			E[%loss] when overhead fee is (in cents)						
c	E[price]	E[rew]	E[loss]	0	5	10	15	20	25	30
3	81.00	72.51	8.48	10.48	15.67	20.31	24.46	28.20	31.58	34.67
4	110.02	105.97	4.05	3.68	7.87	11.71	15.24	18.50	21.52	24.31
5	139.99	137.51	2.48	1.77	5.16	8.32	11.28	14.05	16.66	19.11
6	171.03	170.02	1.01	0.59	3.41	6.08	8.61	11.00	13.27	15.43
7	202.94	201.24	1.71	0.84	3.23	5.50	7.67	9.74	11.72	13.61
8	235.90	235.06	0.83	*0.35	*2.42	*4.40	6.31	8.14	9.90	11.59
9	270.10	265.52	4.58	1.70	3.48	5.21	6.87	8.47	10.02	11.52
10	305.01	299.98	5.03	1.65	3.24	4.77	*6.26	*7.70	*9.10	*10.46
11	341.19	330.00	11.19	3.28	4.68	6.03	7.35	8.64	9.88	11.10

case the percentage loss can be reduced artificially, since the denominator increases by 100 but the numerator remains the same. More extremely, if 1000 tosses are paid for then the percentage loss is driven down to almost zero! Notwithstanding, to compare different values of c, invoking monotonic relation, our adopted definition of expected percentage loss works just fine. For the 100% refund option with an overhead fee of 14 cents or less (details are not shown), the best choice is $c = 8$ (and a very large k), but for a fee of 15 cents or more, it is $c = 10$. As anticipated, as the overhead fee increases, so does the player's percentage loss.

An astute reader can anticipate how our post-conference conversation ended:

> "If not a 100% refund of the price of the unused tosses, what then do you mean by a guaranteed refund?"
>
> "A guaranteed refund means a percentage of the purchase price of the unused tosses will be refunded if the player had bought this option by paying an additional overhead fee at the very outset of the game. For instance, in the original game, the guaranteed refund is 0% for an overhead fee of 0 cents: The gambler gets nothing back on the unused tosses; and pays no extra fee. The Gambling House could offer several options: (1) 50% refund for a fee of 5 cents; (2) 60% refund for a fee of 7 cents; (3) 70% refund for a fee of 10 cents; (4) 80% refund for a fee of 15 cents. In each case, we would ask what is the optimum choice of (c, k)? When we answer these questions, we can determine which of the four offers of guaranteed percentage refund is optimum."

In this paper, I will answer the optimal choices in the modified game that offers a partial refund of unused tosses for a modest fee upfront. A player who was intending to play the original ($c = 6, k = 28$) game, when offered the modified game with partial refund, has some incentive to pay for a few extra tosses at the outset in hope of improving her chance of capturing all $c = 6$ nodes; and yet should she capture them early, she can recover a percentage of her wager. What should be her best choice now? If this offer were available at no overhead fee, the player would lower her expected percentage loss below that in the original game (where it was 9.69%). But the presence of an overhead fee makes it challenging to anticipate the expected percentage loss without studying the process in more details.

Moreover, the offer of a partial refund may cause the player to rethink how many nodes she should set out to capture. This paper is devoted to answering the optimal choice among the four percentage refunds—50%, 60%, 70%, 80%—with associated overhead fees 5, 7, 10, 15 cents, respectively—and the corresponding optimal choice of (c, k).

4. Optimal Games Under Different Percentage Refunds

Suppose that the Gambling House offers the gambler for an upfront payment of 5 cents, a 50% refund on the purchase price of all unused tosses by the time the player captures c nodes. How should the player determine (c, k)? We leave to the reader to check that, as it was in the original game, choosing $c = 1$ or $c = 2$ is not good for the gambler.

Proceeding in a routine manner, for every fixed $3 \leq c \leq 11$, one can simulate the expected loss for various choices of $k \geq c$. However, a smarter search algorithm can be implemented: Sarkar (2020) argued that for each contemplated c, the player is better off choosing an *even* $k \geq c$. Roughly speaking, this is because on the parity dial nodes at an odd distance away from Node 0 have smaller labels compared to nodes at an even distance away. Below we exhibit the simulation results for the choice of $c = 8$, and $50 \leq k \leq 60$, demonstrating that indeed k ought to be chosen an even number because for each odd k, the expected percentage loss is lower at both of its even neighbors.

Table 3: For 50% refund at 5 cents, expected reward and expected loss for $c = 8$ and $50 \leq k \leq 60$, exhibit that even values of k are preferable.

		cents			E[%loss]	
c	k	price	E[rew]	E[loss]	k odd	k even
8	50	255	233.54	21.46		8.42
8	51	256	234.29	21.71	8.48	
8	52	257	235.54	21.46		8.35
8	53	258	236.26	21.74	8.43	
8	54	259	237.55	21.45		8.28
8	55	260	238.25	21.75	8.37	
8	56	261	239.47	21.53		*8.25
8	57	262	239.90	22.10	8.44	
8	58	263	240.82	22.18		8.43
8	59	264	241.64	22.36	8.47	
8	60	265	242.62	22.38		8.44

Moreover, having found the optimal k for a specific c, say k_c, the search for the optimal k for $(c+1)$ can be expedited by taking k even, and not just larger than $(c+1)$ but larger than k_c. Henceforth, for all percentage refund options, we shall only look at even $k > k_{c-1}$ corresponding to each contemplated c. In Tables 4 and 5, for the partial refund options (1)–(4) we document the expected percentage loss corresponding to each $c \in \{3, 4, \ldots, 11\}$ and selected k's that help us determine the optimal (c, k). Finally, using Tables 4 and 5, we choose the best among the four positive refund options (1)–(4).

Table 4: Expected percentage losses for $3 \leq c \leq 11$ and selected even k's determine the optimal (c, k), under refund options (1) and (2). For each refund option, min E[%loss] within c is marked by *, and the minimum overall by #.

		(1) 50% refund at 5 cents						(2) 60% refund at 7 cents			
c	k	price	E[rew]	E[loss]	E[%loss]	c	k	price	E[rew]	E[loss]	E[%loss]
3	8	88	68.88	19.12	21.73	3	12	94	74.92	19.08	20.30
3	10	90	72.14	17.86	19.85	3	14	96	76.75	19.25	*20.05
3	12	92	74.35	17.65	*19.19	3	16	98	78.24	19.76	20.17
3	14	94	75.91	18.09	19.25	3	18	100	79.53	20.47	20.47
3	16	96	77.18	18.82	19.60	3	20	102	80.84	21.16	20.74
4	16	121	105.48	15.52	12.83	4	18	125	108.51	16.49	13.20
4	18	123	107.75	15.25	12.40	4	20	127	110.50	16.50	*12.99
4	20	125	109.53	15.47	*12.38	4	22	129	112.22	16.78	13.01
4	22	127	110.95	16.05	12.64	4	24	131	113.76	17.24	13.16
4	24	129	112.47	16.53	12.81	4	26	133	115.18	17.82	13.40
5	22	152	135.54	16.46	10.83	5	24	156	138.87	17.13	10.98
5	24	154	137.99	16.01	10.39	5	26	158	141.06	16.94	10.72
5	26	156	139.92	16.08	*10.31	5	28	160	143.17	16.83	*10.52
5	28	158	141.62	16.38	10.37	5	30	162	144.88	17.12	10.57
5	30	160	143.32	16.68	10.43	5	32	164	146.44	17.56	10.71
6	32	187	170.11	16.89	9.03	6	34	191	173.54	17.46	9.14
6	34	189	172.28	16.72	* 8.85	6	36	193	175.55	17.45	9.04
6	36	191	174.07	16.93	8.86	6	38	195	177.52	17.48	8.96
6	40	195	177.21	17.79	9.12	6	40	197	179.16	17.84	* 9.06
6	38	193	175.61	17.39	9.01	6	42	199	180.77	18.23	9.16
7	40	220	200.09	19.91	9.05	7	42	224	203.88	20.12	8.98
7	42	222	202.28	19.72	* 8.88	7	44	226	205.81	20.19	8.93
7	44	224	204.09	19.91	8.89	7	46	228	207.88	20.12	* 8.82
7	46	226	205.88	20.12	8.90	7	48	230	209.66	20.34	8.84
7	48	228	207.56	20.44	8.96	7	50	232	211.35	20.65	8.90
8	50	255	233.57	21.43	8.40	8	54	261	239.55	21.45	8.22
8	52	257	235.58	21.42	8.33	8	56	263	241.40	21.60	8.21
8	54	259	237.50	21.50	8.30	8	58	265	243.29	21.71	8.19
8	56	261	239.40	21.60	#* 8.28	8	60	267	245.20	21.80	#* **8.16**
8	58	263	240.85	22.15	8.42	8	62	269	246.81	22.19	8.25
9	60	290	263.10	26.90	9.28	9	66	298	271.11	26.89	9.02
9	62	292	265.04	26.96	9.23	9	68	300	273.11	26.89	* 8.96
9	64	294	266.98	27.02	* 9.19	9	70	302	274.80	27.20	9.01
9	66	296	268.71	27.29	9.22	9	72	304	276.72	27.28	8.97
9	68	298	270.42	27.58	9.26	9	74	306	278.33	27.67	9.04
10	72	327	297.21	29.79	9.11	10	76	333	303.69	29.31	8.80
10	74	329	299.15	29.85	9.07	10	78	335	305.75	29.25	* 8.73
10	76	331	301.05	29.95	* 9.05	10	80	337	307.38	29.62	8.79
10	78	333	302.88	30.12	9.05	10	82	339	309.38	29.62	8.74
10	80	335	304.48	30.52	9.11	10	84	341	311.00	30.00	8.8
11	86	366	328.17	37.83	10.34	11	90	372	335.15	36.85	9.91
11	88	368	330.38	37.62	10.22	11	92	374	336.85	37.15	9.93
11	90	370	332.22	37.78	*10.21	11	94	376	338.76	37.24	9.91
11	92	372	333.74	38.26	10.29	11	96	378	340.62	37.38	* 9.89
11	94	374	335.34	38.66	10.34	11	98	380	342.31	37.69	9.92

Table 5: Expected percentage losses for $3 \leq c \leq 11$ and selected even k's determine the optimal (c, k), under refund options (3) and (4). For each refund option, min E[%loss] within c is marked by *, and the minimum overall by #.

		(3) 70% refund at 10 cents						(4) 80% refund at 15 cents			
c	k	price	E[rew]	E[loss]	E[%loss]	c	k	price	E[rew]	E[loss]	E[%loss]
3	14	99	77.55	21.45	21.67	3	22	112	85.29	26.71	23.85
3	16	101	79.20	21.80	21.59	3	24	114	86.88	27.12	23.79
3	18	103	80.77	22.23	*21.58	3	26	116	88.52	27.48	23.69
3	20	105	82.24	22.76	21.68	3	28	118	90.12	27.88	23.63
3	22	107	83.63	23.37	21.84	3	30	120	91.68	28.32	*23.60
4	20	130	111.54	18.46	14.20	4	26	141	118.38	22.62	16.04
4	22	132	113.54	18.46	*13.99	4	28	143	120.15	22.85	15.98
4	24	134	115.11	18.89	14.10	4	30	145	121.85	23.15	*15.97
4	26	136	116.87	19.13	14.07	4	32	147	123.49	23.51	15.99
4	28	138	118.35	19.65	14.24	4	34	149	125.07	23.93	16.06
5	28	163	144.46	18.54	11.38	5	34	174	151.79	22.21	12.76
5	30	165	146.39	18.61	*11.28	5	36	176	153.71	22.29	12.67
5	32	167	148.13	18.87	11.30	5	38	178	155.50	22.50	*12.64
5	34	169	149.89	19.11	11.31	5	40	180	157.04	22.96	12.75
5	36	171	151.51	19.49	11.40	5	42	182	158.81	23.19	12.74
6	38	198	179.28	18.72	9.45	6	46	211	188.85	22.15	10.50
6	40	200	181.21	18.79	* 9.40	6	48	213	190.64	22.36	10.50
6	42	202	182.94	19.06	9.44	6	50	215	192.51	22.49	*10.46
6	44	204	184.77	19.23	9.43	6	52	217	194.24	22.76	10.49
6	46	206	186.40	19.60	9.51	6	54	219	195.95	23.05	10.53
7	48	233	211.84	21.16	9.08	7	58	248	223.73	24.27	9.79
7	50	235	213.64	21.36	9.09	7	60	250	225.54	24.46	9.79
7	52	237	215.57	21.43	* 9.04	7	62	252	227.48	24.52	* 9.73
7	54	239	217.34	21.66	9.06	7	64	254	229.07	24.93	9.82
7	56	241	219.15	21.85	9.07	7	66	256	230.89	25.11	9.81
8	58	268	245.87	22.13	8.26	8	70	285	260.11	24.89	8.74
8	60	270	247.84	22.16	8.21	8	72	287	262.02	24.98	8.70
8	62	272	249.71	22.29	#* 8.20	8	74	289	263.87	25.13	* 8.69
8	64	274	251.38	22.62	8.19	8	76	291	265.68	25.32	8.70
8	66	276	253.39	22.61	8.26	8	78	293	267.24	25.76	8.76
9	72	307	279.68	27.32	8.90	9	82	322	292.56	29.44	9.14
9	74	309	281.39	27.61	8.94	9	84	324	294.30	29.70	9.17
9	76	311	283.46	27.54	* 8.85	9	86	326	296.18	29.82	* 9.15
9	78	313	285.11	27.89	8.91	9	88	328	297.96	30.04	9.16
9	80	315	286.95	28.05	8.90	9	90	330	299.63	30.37	9.20
10	84	344	314.41	29.59	8.60	10	100	365	333.37	31.63	8.67
10	86	346	316.45	29.55	8.54	10	102	367	335.23	31.77	8.66
10	88	348	318.38	29.62	* 8.51	10	104	369	337.33	31.67	#* 8.58
10	90	350	320.02	29.98	8.57	10	106	371	338.65	32.35	8.72
10	92	352	321.70	30.30	8.61	10	108	373	340.42	32.58	8.73
11	98	383	345.96	37.04	9.67	11	112	402	363.42	38.58	9.60
11	100	385	347.67	37.33	9.70	11	114	404	365.38	38.62	9.56
11	102	387	349.84	37.16	* 9.60	11	116	406	367.17	38.83	* 9.56
11	104	389	351.32	37.68	9.69	11	118	408	368.85	39.15	9.59
11	106	391	353.13	37.87	9.68	11	120	410	370.66	39.34	9.60

From Tables 4 and 5, we note that for the partial refund options of 50%, 60%, 70% and 80% on payment of 5, 7, 10, 15 cents, respectively, the optimal (c, k) are $(8, 56)$, $(8, 60)$ $(8, 62)$ and $(10, 104)$. Recall that in the original game with no refund the optimal game was $(6, 28)$. Thus, as the refund percentage increases, the gambler not only may choose to buy more tosses, but also commit to capturing more nodes! The safety net of getting a partial refund on prepaid excess tosses, makes the player more inclined to targeting a higher c and choosing a higher k.

5. Properties of the Optimal Game Under the Optimal Refund Option

In the previous section, we learned that among the various refund options offered to the gambler, the best is Option (2): 60% refund on payment of 7 cents. For this case, the optimum (c, k) is $(8, 60)$. This optimal game has admission price $25 \times 8 + 60 + 7 = 267$ cents. For this optimum game, we exhibit some characteristics such as the number of tosses until the game ends, the probability distribution of the number of nodes captured, the probability distribution of the farthest node captured going clockwise from Node 0, and the probability distribution of the reward earned (plus refund).

Based on a simulation of 10^6 (one million=ten lakhs) iterations of game $(8, 60)$, we can estimate the number of tosses until the game ends by capturing all 8 nodes using Figure 2.

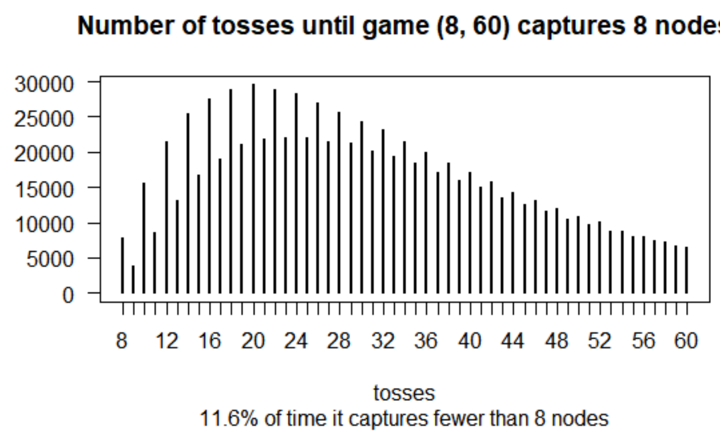

Figure 2: The number of tosses until game $(8, 60)$ ends by capturing 8 nodes. About 11.6% of times the game captures fewer than 8 nodes in 60 tosses.

The probability of capturing 8 nodes in 8, 9 or 10 tosses are respectively $2^{-7}, 2^{-8}, 2^{-6}$. This is supported by the simulation where out of 10^6 iterations the frequencies of 8, 9, 10 tosses are respectively 7790, 3896, 15632 (*P-values* of chi-square tests with one degree of freedom are respectively .8027, .8758, .9582). We leave it to the inquisitive reader to explain a curious feature observed in Figure 2: The relative frequencies for odd number of tosses are smaller when compared to those of their two immediate neighboring even values!

The number of tosses until game (8, 60) ends with 8 nodes captured is shown in Figure 2. In 115,872 more iterations (not shown in Figure 2) fewer than 8 nodes are captured in 60 tosses. For these iterations, how many nodes are actually captured? The answer is given in Table 6, which also shows the corresponding probabilities, which are correct to three decimal places because their 95% confidence intervals are at most 0.001 wide (see Devore and Berk, 2007, for example). All 8 vertices are captured 88.4% of times; 7 nodes 7.2% of times and 6 or fewer nodes 4.4% of times.

Table 6: The simulated distribution of the number of nodes captured by game $(8, 60)$ **shows about 11.6% of time not all 8 nodes are captured.**

# nodes	1	2	3	4	5	6	7	8	sum
frequency	0	0	9	841	8823	34224	71975	884128	1000000
probability	.000000	.000000	.000009	.000841	.008823	.034224	.071975	.884128	1.000000

Using the information in Figure 2 and Table 6, the number of tosses until game (8, 60) ends has the following summary statistics (see R code in the Annexure):

$$N=10^6, \text{ Min}=8, \text{ Q1}=21, \text{ Q2}=31, \text{ Mean}=33.88, \text{ Q3}=46, \text{ Max}=60, \text{ SD}=15.50$$

The probability distribution of the number of tosses left over when the game ends is obtained simply by subtracting from 60 the number of tosses needed to capture 8 nodes (and adding 0.115872 to the probability that no toss is left over). Thereafter, one can construct the probability distribution of the refund amount by multiplying the number of leftover tosses by the refund percentage.

Also, based on this same simulation, and using the built-in kernel density estimator in R (see Silverman, 1986), the estimated probability density function of the reward earned (plus refund) is shown in Figure 3, with its summary statistics given by

$$N=10^6, \text{ Min}=64.4, \text{ Q1}=235.2, \text{ Q2}=248.2, \text{ Mean}=245.2, \text{ Q3}=264.2, \text{ Max}=286.2, \text{ SD}=27.82$$

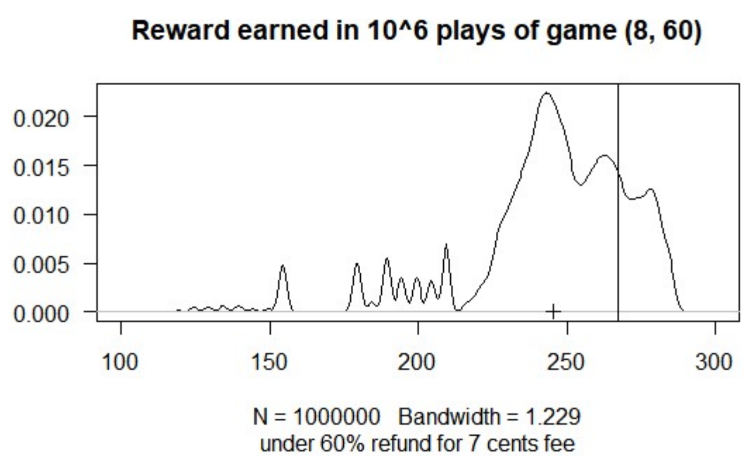

Figure 3: The reward (plus refund) distribution in game $(8, 60)$ **exhibits mean=245.2 (the + sign), SD=27.82, a 20.4% chance of winning (reward** > 267 **cents (the vertical line)), and an expected percentage loss of 8.18.**

Using the estimated density function given in Figure 3, we can infer (see Rohatgi, 2003, for example) some probabilities the gambler would like to know. While on average the gambler loses 8.18% of her wager of 267 cents per game, about 20.4% of the time she earns more reward (plus refund) than the wager. Thus, the game does not always end in a loss for the player. Moreover, a gambler who has a tolerance limit of 10% loss, actually earns back more than 90% of her wager (or 240.3 cents) about 66.1% of times, making the game quite attractive to her!

Finally, note that Option (3), which offers a 70% refund on payment of 10 cents overhead fee, is a close second best when the player chooses ($c = 8, k = 62$), with an expected loss of 8.20%. We leave it to the interested reader to study its properties by adapting the R codes given in the Annexure.

6. Lessons Learnt From The Game

The problem studied in Sarkar (2020) was formulated in response to an invitation to deliver a keynote speech at the 2nd International Conference on "Frontiers of Operations Research & Business Studies" held during 27–28 December 2019, at the Calcutta Business School. The mission of FORBS (see FORBS, 2019) is described as follows:

> Most often organizations are confronted with questions like how to make a good decision? What is really a good decision? What constitutes a poor decision? Is there any pattern in the decisions made? In the quest for finding answers to these questions, the contributions of several disciplines like statistics, mathematics, sociology, economics, information technology, operations research and behavioral science need to be acknowledged. In other words "Decision Sciences incorporate an economic framework—a consistent, rational and objective system to "price" each possible outcome, taking into account risks and rewards."

Sarkar (2020) demonstrated the essential elements of optimal decision making in a rather simple model: Choose (c, k) to minimize the expected percentage loss in the reward-earning binary random walk game on the parity dial. Here we have expanded that problem to incorporate one more layer of complexity: First choose the refund policy offered at several different options with associated overhead fees, and then choose (c, k) to minimize the expected percentage loss.

We showed that when the gambler judiciously buys the optimal option for partial refund, the game may become more favorable to the gambler than playing the original game with no refund. However, we will be remiss if we did not mention that the Gambling House still has the last laugh: It can, for instance, raise the overhead fee for each refund option by, say, 5 cents. Then the best option for the gambler will be the original game with an expected percentage loss of 9.79. The other refund options (1)–(4) has expected percentage loss of 10.00, 9.853, 9.852, 9.805, respectively. To calculate these percentage losses, simply take the ratio of expected loss and price after adding to both quantities the change in the overhead fee, as in Table 2. See R codes in the Annexure. However, to keep the gambler playing, the Gambling House cannot remain totally adversarial; it must keep the overhead price in check. It is precisely this tension that keeps decision making exciting and intriguing.

The reward-earning random walk game, translates in the marketplace into a decision about investing sufficient resources to ensure a good chance of fulfilling the mission of a venture. However, to avoid letting the unused part of the investment go to waste, the entrepreneur will act prudently by purchasing an insurance to protect the resource. Thus, the refund option can be thought of as an insurance policy. Should the entrepreneur accomplish the goal of the venture and still have some resources left over, she will at least get back a predetermined percentage as refund. The insurance company that underwrites such an insurance plan likely has a market where they can resell the leftover resources and pass on (part of) the proceeds to the insuree.

The central message of the reward-earning random walk games is that while facing uncertainty of outcomes, an entrepreneur can and should make the best decision based on the information available, and adjust the decision should the conditions change. A careful and adequate planning and flexibility in decision making are necessary to maximize the expected return from a venture.

Acknowledgment

I thank the participants of the 22nd Annual Conference of the SSCA, especially the discussant during high tea, for their inquisitive questions which inspired me to solve the problem I had presented there as unsolved.

References

Devore, J. L. and Berk, K. N. (2007). *Modern Mathematical Statistics.* Thomson-Brooks / Cole, Belmont, California.

FORBS (2019). *Mission Statement.* https://calcuttabusinessschool.org/FORBS2019/

Lovasz, L. (1993). Random walks on graphs: A survey. In: *Combinatorics, Paul Erdös is Eighty*, Bolyai Society Mathematical Studies, 2 Keszthely (Hungary), 11–46.

Maiti, S. I. and Sarkar, J. (2019). Random walks on paths and cycles. *Mathematics Magazine*, **92(4)**, 252–268. DOI: 10.1080/0025570X.2019.1611166

Medhi, J. (1982). *Stochastic Processes.* A Halsted Press Book, John Wiley & Sons, New York.

Rohatgi, V. K. (2003). *Statistical Inference.* Second Edition, Dover, New York.

Ross, S. M. (1996). *Stochastic Processes.* Second Edition, Wiley Series in Probability and Statistics, John Wiley & Sons, New York.

Sarkar, J. (2006). Random walk on a polygon. In: *Recent Developments in Nonparametric Inference and Probability*, IMS Lecture Notes **50**, 31–43. Institute of Mathematical Statistics, Beachwood, Ohio.

Sarkar, J. (2020). Rules, resolutions, randomness and reward: Lessons from a game. *Proceedings of the 2nd International Conference on "Frontiers of Operations Research & Business Studies" (FORBS)*, 27–28 December 2019, Calcutta Business School.

Silverman, B. W. (1986). *Density Estimation.* Chapman and Hall, London, UK.

ANNEXURE

We document the R codes used to prepare Tables 2–6 and draw Figures 2 and 3.

```
### Table 2: Simulate expected nb of tosses, price reward, loss and %loss
## with 0% refund at 0 cents overhead fee
k=1000  # a very large number of tosses
reward=function(c,k){    # c=vertices to capture, k=tosses allowed
  rf=c(1,11,3,9,5,7,6,8,4,10,2,0)
  ber=2*rbinom(k,1,1/2)-1; cber=c(0,cumsum(ber));
  (nv=cummax(cber)-cummin(cber)); l=sum(nv<c); l1=l+1
    if(l<=k){cber=cber[1:l1]}
  maxv=max(cber); vv=seq(min(cber),max(cber))
  visited=vv*(vv>0)+(vv+12)*(vv<=0);
  nvv=length(visited); rew=5*sum(rf[visited])
  c(nvv,maxv,rew,l) }

el=matrix(0,9,6)   # initialize E[%loss] matrix
for (i in 1:9){
  data=replicate(10^5,reward(i+2,k))
  price=0+25*(i+2) + mean(data[4,])   # no fee yet
  rew=mean(data[3,]);  loss=price-rew
  el[i,]=(c( round(i+2,0), round(k,0), round(price,2),
     round(rew,2),round(loss,2), round(100*loss/price,2) ) }
el # calculate more columns as (E[loss]+fee)/(E[price]+fee)

### Tables 3, 4 and 5: How much is the random reward?
reward=function(c,k){ # c=vertices to capture, k=tosses allowed
  rf=c(1,11,3,9,5,7,6,8,4,10,2,0) # nickels at the nodes
  ber=2*rbinom(k,1,1/2)-1; cber=c(0,cumsum(ber));
  (nv=cummax(cber)-cummin(cber)) # nb of non-zero vertices
  l=sum(nv<c)  # nb tosses until capture c nodes
  l1=l+1; if(l<=k){cber=cber[1:l1]} # cber has an initial 0
  maxv=max(cber); vv=seq(min(cber),max(cber)) # vertices captured
  visited=vv*(vv>0)+(vv+12)*(vv<=0) # recode vertices captured
  nvv=length(visited) # nb of nodes visited (includes Node 0)
  rew=5*sum(rf[visited]) + (k-l)*0.60 # add refund (% of excess tosses)
  c(nvv-1,maxv,rew,l) } # outputs

## (refund %, payment)=(.50, 5), (.60, 7), (.70, 10), (.80, 15)
## simulate expected nb of tosses, price, reward, loss and %loss
c=8  # 2, 3, 4, ..., 11
for (k in seq(58,62,2)){     # try a range of values of k (even)
  price=10+25*c+k  # overhead fee + admission
  data=replicate(10^5,reward(c,k))
  rew=mean(data[3,]); loss=price-rew
```

```
  print(c( round(c,0), round(k,0), round(price,2),
   round(rew,2),round(loss,2), round(100*loss/price,2) )) }

### Figures 2, 3 and Table 6: Properties of the optimum game
##  (refund 60%, overhead 7 cents) (c=8, k=60)
c=8; k=60
data=replicate(10^6,reward(c,k))
summary(data[4,])   # nb of tosses=61 means < 8 nodes captured

# Figure 2.
plot(table(data[4,])[1:53], las=1, ylab='', xlab='tosses',
   main="Number of tosses until game (8, 60) captures 8 nodes",
   sub="11.6% of time 60 tosses capture fewer than 8 nodes")
prop.test(7790,10^6,1/128) # test P{ntoss=8)=2/2^8=1/2^7
prop.test(3896,10^6,1/256) # test P{ntoss=9)=2/2^9=1/2^8
prop.test(15632,10^6,1/64) # test P{ntoss=10)=2*8/2^10=1/2^6
nbtoss=data[4,]-(data[4,]==61) # nb tosses at game end (60 replaces 61)
summary(nbtoss); sd(nbtoss)

# Table 6.
summary(data[1,]); table(data[1,]) # nb of nodes captured

# Calculate E[%loss]
summary(data[3,]) # reward earned (plus refund)
price=25*c+k+7    # include overhead fee
rew=mean(data[3,]); loss=price-rew
print(c( round(c,0), round(k,0), round(price,2),
   round(rew,2),round(loss,2), round(100*loss/price,2) ))

# Figure 3. Kernel Density Plot
d <- density(data[3,])   # returns density data
plot(d, las=1, xlim=c(100,300), ylab='',
   main="Reward earned in 10^6 plays of game (8, 60)",
   sub="under 60% refund for 7 cents fee; price=267") # plots the results
abline(v=267); points(245.2,0, pch=3)   # reference price and mean reward

sum(data[3,]>price)/10^6 # prob of winning above the price
sum(data[3,]>0.90*price)/10^6 # prob of earning above the 10% loss threshold

### What if the overhead fees change?
# Simply revise the expected % loss
eloss=c(21.60, 21.80, 22.29, 31.67)
price=c(261, 267, 272, 369)
for (i in -5:8){print((eloss+i)/(price+i))}
```