e-Research: A Journal of Undergraduate Work

Volume 2
Number 2 *Vol 2, No 2 (2011)*

Article 5

September 2014

A Two-Light Version of the Classical Hundred Prisoners and a Light Bulb Problem: Optimizing Experimental Design through Simulations

Alexander S. Barrett

Cyril Rakovski

e-Research: A Journal of Undergraduate Work, Vol 2, No 2 (2011)

A Two-Light Version of the Classical Hundred Prisoners and a Light Bulb Problem: Optimizing Experimental Design through Simulations

Alexander S. Barrett, Cyril S. Rakovski

Abstract

We propose five original strategies of successively increasing complexity and efficiency that address a novel version of a classical mathematical problem that, in essence, focuses on the determination of an optimal protocol for exchanging limited amounts of information among a group of subjects with various prerogatives. The inherent intricacy of the problemsolving protocols eliminates the possibility to attain an analytical solution. Therefore, we implemented a large-scale simulation study to exhaustively search through an extensive list of competing algorithms associated with the above-mentioned 5 generally defined protocols. Our results show that the consecutive improvements in the average amount of time necessary for the strategy-specific problem-solving completion over the previous simpler and less advantageously structured designs were 18, 30, 12, and 9% respectively. The optimal multi-stage information exchange strategy allows for a successful execution of the task of interest in 1722 days (4.7 years) on average with standard deviation of 385 days. The execution of this protocol took as few as 1004 and as many as 4965 with median of 1616 days.

Keywords: Large Scale Simulations, Classical Mathematics Puzzles, 100 prisoners and light bulb puzzle

Introduction

Conceptually, this research addresses a special case of the important problem of finding optimal design for dissemination of information among the subject of a population of interest under severe restrictive rules that include very limited amount of information allowed to be transferred at each step of the algorithm. In particular, we studied a novel version of the hundred prisoners and a light bulb classical mathematical puzzle, defined as follows:

One hundred prisoners have been newly ushered into prison. The warden tells them that starting tomorrow, each of them will be placed in an isolated cell, unable to communicate amongst each other. Each day, the warden will choose one of the prisoners uniformly at random with replacement and place him in a central interrogation room containing only a light bulb with a toggle switch. The prisoner will be able to observe the current state of the light bulb. If he wishes, he can toggle the light switch. He also has the option of announcing that he believes all prisoners have visited the interrogation room at some point in time. If this announcement is true, all prisoners are set free, but if it is false, all prisoners are executed. The warden leaves and the prisoners huddle together to discuss their fate.

Can they agree on a protocol that will guarantee their freedom in the shortest amount of time?

Clearly, the switch can display only two states (say 0 and 1) and at each distinct step of a successful scheme, only two different state transitions are possible, the single switch position being changed from 0 to 1 or vice versa. Extensive study of protocols and their properties has already been accomplished (Wu, 2002), (Wu, Wu riddles, 2002). Moreover, previous research work has proposed and discussed several variations and generalizations of this classical problem (Paul-Olivier Dehaye et al., 2003).

In this study, we investigate the same problem with the novel assumption that two light bulbs with two separate switches are present in the interrogation room which allows us to use combinations of switch positions to display four states (say 0, 1, 2, and 3 expressed by the binary representations 00, 01, 10 and 11 respectively) and thus, at each step of a successful protocol, twelve different state transitions are possible. The new two-light bulb setting allows for the design of several influential new approaches for strategy optimization that combined with the range of time points assigned to their actual implementation, result in a great variety of potential best protocols. We empirically estimated the distribution of the number of the days until the successful completion of each competing strategy through large-scale simulations due to their inherent analytically intractable complex behavior. All calculations were carried out using the R statistical package (R Development Core Team, 2011) (http://www.r-project.org).

Methods and Results

In the subsequent presentation we describe and analyze the performance of five protocols of increasing complexity and efficiency. The first two strategies have direct counterpart designs in the one light bulb setting which allows us to also assess the improvement in the time until their successful completion directly attributable to the additional information transferable via the second light. In the successive sections we use the term strategy, design, algorithm, scheme and protocol interchangeably. In all protocols, we conceptualize the problem setting as initially assigning a single token to each prisoner representing the fact that they have not been counted as subjects that have visited the interrogation room.

Prisoners can deposit their token by increasing the value of the switch in certain design rounds or by not interacting with the switch in other stages. The collection of the tokens is assigned to one or more counters that can manipulate the state of the switch in order to accumulate predetermined quotas or deposit these quotas for future collection by a master counter. Depending on the strategy scheme and stage, prisoners can decrease the value of the switch (consequently adopting extra tokens) in an effort to extend the continuity of a current effective algorithm step. When multiple courses of action are possible, we give priority to single tokens over quota depositions being displayed on the switch as usually more than one counter can collect the single token but only one master counter can retrieve the quota from the switch. Lastly, in multi-stage algorithms, the prisoner arriving on the first day of an adjacent stage, manipulates the switch and his own count of single tokens as a technique that permits a correct transition between stages during which the switch states represent different quantities.

1. Single Preselected Counter Protocol

This simple protocol preselects a leader who is responsible for collecting the tokens of the rest of the prisoners (drones). More formally, let SW denote the value represented by the combination of the two switches. Then,

a) If a drone enters and has a token and SW<3, he deposits his token by incrementing the SW by 1.
b)If the leader enters, he collects the number of tokens represented by the SW and adds it to his count. He then sets the SW to 0.
c)The leader declares victory when the number of collected tokens reaches 100.

This protocol does not depend on any parameters. We simulated 100,000 implementations of this strategy (summary statistics presented in Table 1). Our results show that using this design, the average number of days until the leader declares victory is 3722 (an improvement of 6696 days or 64% over the comparable protocol with one light) with standard deviation of 581. The execution of protocol 1 took as few as 1786 and as many as 6848 with median of 3690 days.

2. Single Dynamically Selected Counter Protocol

Similar to the one switch protocols, we can improve on the two-switch Single Counter Protocol by choosing to elect a leader during a special snowball strategy round rather than preselecting him. In contrast to the previous protocol where the switch counts new visitors, during the snowball round, the switch is incremented by 1 every time a repeated visitor enters the interrogation room. The leader is selected to be the first person to set the switch to 3. The efficiency benefit of this approach is underlined by the fact that during the first 71 days, the probability of selecting a new prisoner is greater than the probability of selecting a repeated visitor (based on simulation study not shown here). Thus, it is advantageous to count the less likely event of repeated visitors and not maximize the value on the switch quickly. Once the leader has been selected and the snowball round of predetermined length has finished executing, we continue the protocol with subsequent stage identical to protocol 1. More formally,

Stage 1 (snowball round) of length n days:
For days 1 through n-1, if SW<3:
If the prisoner who enters has a token, he leaves the SW as is, effectively depositing token.
If the prisoner who enters does not have a token, he increments the SW by 1. If this would make SW=3, this prisoner becomes the leader and initializes his count to k-3, where k is current day of the snowball round.
If SW=3, whoever enters does nothing.

For day n:
If SW < 3, whoever enters becomes the leader, setting his count at n-SW.
If SW = 3, whoever enters does nothing.
SW is now set to 0 for round 2.

Stage 2:
Begins at day n+1 and continues as the normal Single Counter Protocol does.

This protocol depends on a single parameter that denotes the length of the snowball round. We simulated 100,000 implementations of this strategy for snowball round lengths ranging from 20 to 40 days. This range is selected based on the distribution of the number of of days necessary for the switch to fill up. Based on a simulation study not presented here, we concluded that the duration of the snowball round that minimizes the time until a successful completion of the protocol is 35. Our results show that using optimal version of this design, the average number of days until the leader declares victory is 3039 (an improvement of 6264 days or 67% over the comparable protocol with one light and an improvement of 683 days or 18% over protocol 1) with standard deviation of 557. The execution of protocol 2 took as few as 1161 and as many as 6531 with median of 3004 days (summary statistics presented in Table 1).

Table 1. Summary statistics of the results for protocols 1 and 2 *.

Protocol	Mean	Median	SD	Min	Max
1	3722	3690	581	1786	6848
2	3039	3004	557	1161	6531

*Based on 100,000 simulation samples per scenario.

3. Multiple Preselected Counters Protocol

The inherent flaw of protocol 1 is that often the switch is loaded and awaiting the arrival of the leader who enters the interrogation room every 100 days on average and during that time the new visitors cannot deposit their tokens. As discussed in Wus paper 100 Prisoners and a Light Bulb, an improvement can be attained by preselecting m assistant counters (or ACs) who collect the single tokens until they reach predetermined quotas. Once they reach their quotas, they can deposit them for future collection by the leader. Therefore, we reserve one of the switch states to denote a single quota, i.e. in this protocol the assigned switch values are as follows, SW=(0 tokens, 1 token, 2 tokens, 1 quota). The leader declares victory as soon as he collects all m quotas from the ACs.

This protocol depends on a single parameter that denotes the number of ACs. We simulated 100,000 implementations of this strategy for 5, 6, 7, 8, 9, and 10 assistant counters. Our results show that the optimal version of this design is attained by 8 counters and the corresponding average number of days until the leader declares victory is 2123 (an improvement of 916 days or 30% over the fastest variant of protocol 2) with standard deviation of 353. The execution of protocol 3 took as few as 985 and as many as 4271 with median of 2093 days (summary statistics presented in Table 2).

Table 2. Summary statistics of the results for protocol 3[*].

Number of Counters	Mean	Median	SD	Min	Max
5	2187	2157	344	1134	4299
6	2187	2157	344	1059	4185
7	2190	2159	344	1199	4189
8	**2123**	**2093**	**353**	**985**	**4271**
9	2147	2117	361	1040	4363
10	2146	2116	361	1010	4504

[*]Based on 100,000 simulation samples per scenario.

4. Multiple Dynamically Selected Counters Protocol

In essence, this protocol combines and exploits the advantageous properties of strategies 2 and 3. This scheme starts with *m* consecutive snowball rounds designed to dynamically elect *m* counters (the leader and the remaining *m*-1 assistant counters). After the end of all snowball rounds, the *m* counters have been selected and the protocol continues with stage 2 in a fashion identical to that of protocol 4. As mentioned earlier, based on an additional simulation study, during the first 71 days, the probability of selecting a new prisoner is greater than the probability of selecting a repeated visitor which shows that the successive snowball rounds should extend around 71 days in total. Further, we completed a separate simulation study (results not presented) to assess the average number of days necessary for the switch to consecutively fill up. Based on these two ancillary results, for 7, 6, 5 and 4 snowball rounds, we define the following snowball round lengths (26,11,9,8,7,7,6), (28,12,10,9,8,7), (30,14,11,10,9) and (33,16, 13,13) respectively. Lastly, we determine and assign the corresponding quotas to the counters by uniformly distributing the expected number of uncollected tokens which produces the following values (29,15,13,12,11,11,9), (30,17,15,14,12,12), (33,20,17,16,14) and (37,24,20,19).

This protocol depends on a several parameters, number of snowball rounds, snowball rounds lengths, and quota values. In this analysis, we simulated 100,000 implementations of this strategy for 4, 5, 6 and 7 snowball rounds and fixed the corresponding lengths and quotas to the values determined by the above-mentioned arguments. Our results show that the optimal version of this design is attained for 6 snowball rounds and the corresponding average number of days until the leader declares victory is 1871 (an improvement of 252 days or 12% over the fastest variant of protocol 3) with standard deviation of 374. The execution of protocol 4 took as few as 816 and as many as 4727 with median of 1871 days (summary statistics presented in Table 3).

Table 3. Summary statistics of the results for protocol 4*.

Number of Counters	Mean	Median	SD	Min	Max
4	1986	1935	407	899	4772
5	1899	1855	386	777	4483
6	**1871**	**1829**	**374**	**816**	**4727**
7	1884	1845	373	888	4256

*Based on 100,000 simulation samples per scenario.

5. Multiple Dynamically Selected Counters Combined with Three Additional Stages of Distinct Switch Allocation States Protocol

This protocol is a multi-parameter and multi-stage advancement of protocol 4 for which we optimize the intricate interplay of all strategy-defining variables. We begin the algorithm with a fixed number of snowball rounds identical to stage 1 of design 4. Next, we incorporate a stage 2, characterized with SW=(0 tokens, 1 token, 2 tokens, 3 tokens), which allows us to fully exploit the switch capacity during the period when no quotas have likely been reached. Further, we integrate a stage 3, characterized with SW=(0 tokens, 1 token, 2 tokens, 1 quota), which allows the deposit of the first available quotas while also permitting for the switch to collect the likely 1 or 2 single tokens for collection by the counters. Lastly, we incorporate a stage 4, characterized with SW=(0 tokens, 1 token, 1 quota, 2 quotas), which allows the deposit of the likely already accumulated 1 and 2 quotas while also permitting for the switch to collect the last remaining single tokens. Evidently, this protocol depends on the following set of parameters, number of snowball rounds (N), duration of all snowball rounds ($S=(s_1, s_2, , s_N)$), quotas assigned to each counter ($Q=(q_1, q_2, , q_N)$), duration of stage 2 ($D2$), and duration of stage 3 ($D3$). We fixed the quotas to the values determined via the argument presented in protocol 4 and created a grid of parameter values consisting of all possible 1024 combinations of the following quantities, 4, 5, 6 and 7 snowball rounds, 16 vectors of snowball lengths that cover sets of values centered about the quantities argued in protocol 4, 200, 400, 600 and 800 days for stage 2 lengths, 100, 300, 500, and 700 days for stage 3 lengths. We simulated 10,000 implementations for this strategy for each of the parameter combination defined above (summary statistics presented in Table 4). Interestingly, the top 25 best performing protocol variants were derivations of the 6 snowball round strategy. Our results show that the optimal design is given by N=6, S=(26,10,8,7,6,5), Q=(31,17,15,14,12,11), D2=800, D3=300. This best performing version of scheme 5 attained the lowest average number of days until the leader declared victory of 1706 (an improvement of 165 days or 9% over the fastest variant of protocol 4) with standard deviation of 385. The execution of protocol 5 took as few as 1004 and as many as 4965 with median of 1616 days.

Table 4. Summary statistics of the results for the top 3 (out of the analyzed 1024) best performing versions of protocol 5*.

Combination of number of counters (N), lengths of snowball periods (S), quotas (Q), length of stage 2 (D2), length of stage 3 (D3)	Mean	Median	SD	Min	Max
N=6, S=(26,10,8,7,6,5), D2=800, D3=300	1706	1616	385	1004	4965
N=6, S=(26,10,9,8,7,7), D2=800, D3=300	1710	1617	387	1041	5308
N=6, S=(25,10,8,8,7,7), D2=800, D3=300	1711	1617	392	1062	4453

*Based on 10,000 simulation samples per scenario.

Discussion

We have performed a large scale simulation study to investigate the performance of thousands of parameter-based variations of 5 competing protocols that successively incorporate various advantageous techniques and attain progressively increasing design sophistication and performance. The average times required for these strategies to complete the task of interest were 3722, 3039, 2123, 1871 and 1706 days with corresponding consecutive improvements of 18, 30, 12, and 9% respectively. However, we realize that the optimal strategy that emerged from our analysis is not the most efficient one. In the subsequent discussion we outline several algorithms and design steps that will potentially decrease the number of days until victory is declared.

We can instruct a dynamically selected counter that finds the SW nonempty at a later snowball round, to reduce both his token count and the SW value by the same number in an effort to permit the current efficient collection of tokens to continue. In particular, if the switch is to become full (which would force him to be elected as a counter again), the subtraction of tokens from his count obtained in a previous snowball round would allow the current snowball round and associated search for a counter to continue. Further, if a dynamically selected counter is to become elected for a second time on the last day of another snowball period and SW<3, he can increment the SW by one to inform the prisoner at the start of the next snowball (or the first prisoner of Stage 2 if this is the last snowball phase) that they are responsible for the quota and the count represented by the SW. This process can be extended 3 days into the next round (by assigning SW values exceeding the values possible under the assumption of a normal counter selection in the previous snowball round) in an effort to reduce the probability of reselecting counters (which dramatically increases the completion time of all protocols).

Although it seems obvious that a counter should implement the first strategy outlined above, it is not clear during which part of the snowball round this approach is really beneficial or exactly how many tokens he should reduce his count and the SW by. For instance, the adoption of this approach seems insightful if a previously elected counter enters very early on in a snowball round and finds SW= 2 as there is a great chance that the switch will become full long before the end of the snowball round, thus resulting in many unproductive days for the remainder of the round during which no one can do anything. However, if it is in the last couple of days of a snowball round when this counter enters and SW=1, there is much less danger of the SW becoming full, and even if it does, there are very few unproductive days left. Thus, there appears to be little upside for the implementation of the reduction strategy in this case. In fact, it is entirely possible that there is even a downside for such decision, as it would likely detrimentally affect the efficient execution of stage 2 which depends on the presence of approximately identical differences between all counters counts and quotas. If these differences are imbalanced, some counters will meet their quotas too early and therefore relinquish themselves of the capability to collect tokens and free up the switches for uninterrupted, continuous use. This means that a counter should be conscious and make decisions based on difference of his quota and count. He would only have true information about his own count and would have to use the expected counts for other snowball rounds given his observed information. Thus, he should manipulate the SW to assure to the best of his knowledge that all counters have similar differences between their quotas and counts. The number of tokens a returning counter should deposit in a snowball round when SW<3 is another parameter that should be varied and examined through simulations in order to best obtain a uniformly optimal protocol.

If a counter elected in an earlier snowball round comes in to find the SW full, he can set the SW to 0, 1, or 2, effectively depositing k SW of his tokens where k is the number of days since the start of the current snowball round. For the remainder of the snowball round, everyone who enters will have assumed that no counter has been elected for the round and act accordingly. This essentially allows a counter elected from a previous snowball to transfer his responsibility to someone else in a later snowball round all the while not interfering with that round standard counter selection. However, because everyone for the remainder of the snowball will have no way of differentiating this case from normality, the counter elected from the previous snowball can only pass his responsibility to this round if his quota and the quota of the current snowball are the same. Also, since this counter is essentially relinquishing himself of all his responsibilities to his quota, he must make sure that k SW is less than or equal to his count. Otherwise, he will be left with a negative count and no quota and consequently no means to convey to everyone else that he has a negative count which might produce a false positive victory. Further, if k SW is less than his count, it might not necessarily be advantageous for him to pass his quota depending on how many remaining tokens he is left holding as he will have to enter the room a number of times just to deposit his remaining tokens. If k SW is equal to his count, then it is definitely worth for him to pass the quota, because doing this allows the collection of more tokens in the current snowball round to continue with no down side. Lastly, the leader cannot pass his quota, because the person receiving the quota will have no idea that he is also being transferred the leaders obligation.

There is also another snowball technique that is not available to the one switch schemes. It is safe to say that near the end of a single snowball round, it is likely that SW>0. This means that at a certain point, it would be more practical to reallocate the state of 0 from meaning that 0 repeated visitors have entered the interrogation room to meaning that 3 repeated visitors have done so. If however, we arrive at this point and SW=0 (representing 0 repeats), whoever enters can simply pretend he does not have a token and increment the SW by 1 so that during the next phase of the snowball round, the prisoners do not think that the SW is representing 3 repeated visitors. This concept of reallocating permutations of the SW states during a snowball round can actually be done multiple times. This technique will allow us to extend the duration of each individual snowball round before the SW will become full. The optimal positions of the reallocation points of the SW states needs to be studied in future endeavors.

As discusses earlier, the traditional snowballs are no longer practical when extended beyond day 73, as the probability of selecting new prisoners becomes smaller than that of choosing repeated visitors. Thus, we can prolong the assistant counters selection in reverse snowball rounds during which we count the new visitors instead of repeated ones in a fashion similar to stage 2 of protocol 5, i.e. SW=(0 tokens, 1 token, 2 tokens, 3 tokens). Further, we can split the round into several phases comparable to the technique used in traditional snowball rounds with reallocation points. The end of each phase of a reverse snowball round should be assigned on a day by which the SW becomes full with high probability. If the SW is full by the end of a given phase, then the SW is reset to 0 and all tokens conveyed by the SW are deposited for collection by the counter. This means that everyone in the next phase would know that there are 3 tokens in the bank as well as what the SW is showing on the day they are summoned to the interrogation room. This design allows for a gradual accumulation of tokens eventually collected by the AC at the end of the entire round. However, if the SW is not full and not empty at the end of a phase, the prisoner entering on the last day of the phase can decide to relay the current SW state to the next phase in an effort to borrow tokens form the future. The prisoner that is forced to become the counter due to appropriate SW and time combinations, adjusts his count to match the sum of the current SW value and the total number of deposited tokens from the successfully completed previous phases and sets the SW to a fail setting (say 3) to convey to the prisoners entering in future phases that the AC for the round has already been elected.

Also, as mentioned earlier, previously elected ACs from traditional or reverse snowball rounds can manipulate the switches by depositing tokens of their own to help prolong the productivity of the currently executing round. Lastly, it is obvious that at some point these types of snowballs will become impractical and we will need to set the starting point for Stage 2 where ACs and the leader accumulate and deposit their quotas.

As evidenced by the methods, results and discussion presented, the two-light setting of the hundred prisoners problem includes multifarious design options that encompasses a large number of protocol (and corresponding varieties) inducing parameter combinations. Large-scale future studies would enable us to advance our understanding of the complex interdependences among the strategy defining parameters described above.

References

Wu, W. (2002). 100 Prosoner and a light bulb. *Unpublushed*. http://www.ocf.berkeley.edu/~wwu.

Wu, W. (2002). Retrieved from Wu riddles: http://www.wuriddles.com.

Paul-Olivier Dehaye, D. F. (2003). One hundred prisoners and a light bulb. *Mathematical Intelligencer, 24*(4), 53-61.

R Development Core Team. (2011). *R: A Language and Environment for Statistical Computing.* R Foundations for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.