International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:12, 2009

The Locker Problem with Empty Lockers

David Avis, Luc Devroye and Kazuo Iwama

Abstract—We consider a cooperative game played by n players against a referee. The players names are randomly distributed among n lockers, with one name per locker. Each player can open up to half the lockers and each player must find his name. Once the game starts the players may not communicate. It has been previously shown that, quite surprisingly, an optimal strategy exists for which the success probability is never worse than $1 - \ln 2 \approx 0.306$. In this paper we consider an extension where the number of lockers is greater than the number of players, so that some lockers are empty. We show that the players may still win with positive probability even if there are a constant k number of empty lockers. We show that for each fixed probability p, there is a constant c so that the players can win with probability at least p if they are allowed to open cn lockers.

Keywords—Locker problem, pointer-following algorithms.

I. Introduction

The locker problem is a cooperative game between a team of n players numbered $1, 2, \ldots, n$ and a referee. In the initial phase of the game, the referee randomly places numbers $1, 2, \ldots, n$, one number per locker, in a locker room containing n closed lockers. Each player enters the room in turn and is allowed to view the contents of $n/2$ lockers, one at a time. He wins if he finds his number in one of the lockers. The players may not communicate after the game starts. The team of n players wins if all individual players win. We would like to know what is the best strategy for the team of n players. The locker problem was originally considered by Peter Bro Miltersen, see [3] and [4].

If each player independently chooses $n/2$ lockers to open, he wins with probability $1/2$, hence, by independence, the team wins with probability $1/2^n$. However, using a pointer-following strategy credited to Sven Skylum, that we describe in the next section, the players can win with probablility of about .31 independent of n. The optimality of this strategy was shown by Curtin and Warshauer [2], and we follow their presentation in this paper. Many variations have been proposed [5], and Avis and Broadbent [1] have recently considered a quantum version of the locker problem.

We consider the extension of the locker problem to the case where there are more lockers than players and hence some (say, k) lockers are "empty." This model appeared in [3] in a different context, and the authors conjectured that the winning probability is exponentially small if k is as large as a linear fraction of n. However nothing is known about the concrete winning probability of this model for any value of k. In this paper, we show that if k is fixed then the pointer-following strategy is still powerful, showing that extensions of the strategy can win the game with high probability if the players are allowed to open a constant fraction of the lockers.

David Avis and Luc Devroye are with the School of Computer Science, McGill University
Kazuo Iwama is with the Graduate School of Informatics, Kyoto University

II. Pointer-following algorithms

For fixed integers n, k, suppose we have n players and $n+k$ lockers, k of which do not contain labels. When $k = 0$ the players best strategy is to use the pointer-following algorithm, due to Skylum. Each player i opens locker i and finds a label l. If $l = i$ he stops, otherwise he opens locker l and repeats the process. Eventually each player must find his label. Let $N(i)$ denote the number of lockers opened by player i, and let $N = max_{i=1,2,\ldots,n} N(i)$. The success probablility of this strategy is given by the following theorem. Let $H_n = 1 + 1/2 + ... + 1/n$ be the n-th harmonic number.

Theorem II.1. *(See [2]) When there are no empty lockers,*

$$Pr(N \leq \lceil n/2 \rceil) = H_n - H_{\lceil n/2 \rceil} > 1 - \ln\ 2\ >\ 0.306$$

It is clear that the players success probability will increase if we allow them to open more lockers. In fact, for any fixed probability p there is a constant c such that they can win with probability at least p if they can open at least $\lceil cn \rceil$ lockers. In the next section we state and prove a more general result as Proposition IV.1.

The main goal of this paper is to show that this result can be obtained even if there are a fixed number of empty lockers. Note that the problem is not so trivial already when $k = 1$. Consider for example the rule that a player jumps to a random locker when he encounters the empty locker. Then one can see easily that this rule brings us back to the almost-zero winning probability.

Initially consider the following naive algorithm. The players prepare a random string of length $n + k$, which is common to all the players, consisting of elements from $\{n + 1, n + 2, ..., n + k\}$. Now the players follow the pointer-following algorithm described above. If a player opens locker i which contains no label, he goes to the locker which is in the i-th position in the random string. If we are lucky, all the empty lockers are filled with different numbers from $n + 1$ through $n + k$ and so we obtain a random permutation of $1, 2, ...n + k$. Therefore we are back to the original game and the players win with probability .306 if they are allowed to open about half of the lockers.

However, if two or more empty lockers are filled with the same number, then some players will never be able to find his number by the pointer following and the algorithm fails. The probability that all the empty lockers get different numbers is $k!/k^k$, which is extremely low even for small k. Therefore the above goal will not be met by this naive algorithm.

III. New Algorithms

We now give modifications to the pointer-following algorithm to handle the case where k lockers are empty. The first algorithm uses the observation that there will be no labels

$n+1, ..., n+k$. So a player is free to assign these labels as he opens empty lockers.

Algorithm PF-1

1) Set j=0. Set the current locker for player i to be i.
2) If the current locker is empty he sets $l = n + j$ and increments j, otherwise he observs its label l.
3) If $l = i$ the player terminates, otherwise the current locker is l and he goes back to step 2.

In the next example, and the examples that follow, the label "0" denotes an empty locker.

Example III.1. *Algorithm PF-1:* $n = 9, k = 2$
Lockers contain the labels: 4 5 0 3 2 0 1 6 9 8 7
Player 1 opens lockers: 1 4 3 10 8 6 11 7
Player 2 opens lockers: 2 5
Players 3,4 open same lockers as player 1
Player 5 opens same lockers as player 2
Player 6 opens lockers 6 10 8
Player 7 opens same lockers as player 1
Player 8 opens same lockers as player 6
Player 9 opens locker 9
The players need to open at most 8 lockers to find all the labels, and so $N = 8$.

Algorithm PF-1 is identical to the original pointer-following algorithm when $k = 0$. For any $k \geq 1$ each player must always find his label because the empty lockers are assigned values to complete a permutation of the integers $1, 2, ..., n+k$. For $k = 1$ this completion is unique, as the empty locker is simply assigned label $n + 1$. However if $k \geq 2$ the first empty locker opened will depend the player's number, unless the empty lockers happen to be $n+1, ..., n+k$. Therefore the permutations arrived at by different players may be different.

A second strategy is to reduce the number of empty lockers by sequential search. All players open the lockers sequentially in the same order $n + k, n + k - 1, n + k - 2, ...$ until either they find their label or some predetermined number t of empty lockers remain. The players who have not yet found their label then use algorithm PF-1.

Algorithm PF-2(t)

1) Each player i opens lockers in the order $n + k, n + k - 1, n + k - 2, ...$ until either he finds his label or finds the $(k - t)$-th empty locker.
2) Each player finding his label leaves the game. Let the $(k - t)$-th empty locker be locker $s + 1$.
3) The s remaining players $p(1) < p(2) < ... < p(s)$ renumber themselves $1, 2, ..., s$ respectively. They use algorithm PF-1 with parameters $n = s$ and $k = t$. If the label l is found, they go next to locker $p^{-1}(l)$.

Note that Algorithm PF-2(k) is identical to Algorithm PF-1. We remark that the players do not need to communicate in order to perform Step 3. All remaining players have the same information, and in particular, know that they are the players labelled $p(1), p(2), ..., p(s)$.

Example III.2. *Algorithm PF-2(t):* $n = 9, k = 2, t = 0$
Lockers contain the labels: 3 2 4 5 8 6 0 1 0 9 7
In step (1) 5 lockers are opened by players 2 3 4 5 6 8.
Players 1,7,9 find their labels.
The 6 remaining players relabel themselves 1 2 3 4 5 6 and use Algorithm PF-1 on sequence 2 1 3 4 6 5
Players 1,2,5,6 open two additional lockers.
Players 3 and 4 open one additional locker.
$N = 7$.

Example III.3. *Algorithm PF-2(t):* $n = 9, k = 2, t = 1$
Lockers contain the labels: 3 2 4 5 8 6 0 1 0 9 7
In step (1) 3 lockers are opened by players 1 2 3 4 5 6 8.
Players 7,9 find their labels.
The 7 remaining players relabel themselves 1 2 3 4 5 6 7 and use Algorithm PF-1 on sequence 3 2 4 5 7 6 0 1
Players 1,3,4,5,7 open 5 additional lockers.
Players 2 and 6 open one additional locker each.
$N = 8$.

On the example, Algorithm PF-2(0) did better than and PF-2(1). However this ranking differs depending on the input permutation, We investigate the relationship between the algorithms later in the paper, and see that in fact, PF-2(0) gives the worst winning probability for the players. They do best with PF-1.

IV. Analysis of Pointer-Following Algorithms

In this section we give an analysis of Algorithm PF-1 and Algorithm PF-2(t) for $t = 0, 1$. First we generalize Theorem II.1 to the case where one locker can be empty, and where players are able to open some constant fraction of lockers. If the fraction is large enough, the players can win with arbitrary high probability.

Proposition IV.1. *Let* $k = 0$ *or* 1 *and let* $N(i)$ *denote the number of lockers opened by player* i *in Algorithm PF-1. Let* $N = max_{i=1,2,...,n} N(i)$. *For any fixed probability* $p_1 < 1$, *there is a positive constant* $c_1 = e^{p_1 - 1} < 1$ *such that*

$$Pr(N \leq \lceil c_1(n + k) \rceil) \geq p_1.$$

Proof: The proof uses the same technique as Theorem II.1 [2]. We first prove the proposition for the case $k = 0$. It is well known, and easy to prove, that the probability that a random permutation contains a cycle of length $n/2 < m \leq n$ is $1/m$. Let X be the length of the longest cycle in a random permutation, and let $1/2 \leq c_1 < 1$. Then since a random permutation can contain at most one cycle of length more than $n/2$ we have

$$Pr(X > \lceil c_1 n \rceil) = \sum_{m=\lceil c_1 n \rceil + 1}^{n} \frac{1}{m} = H_n - H_{\lceil c_1 n \rceil}$$

$$\leq \ln\ n - \ln\ c_1 n = \ln\ \frac{1}{c_1}$$

Suppose without loss of generality that $p_1 \geq 1 - \ln\ 2$ and set $c_1 = e^{p_1 - 1}$. It follows that

$$Pr(X > \lceil c_1 n \rceil) \leq \ln\ \frac{1}{c_1} = 1 - p_1.$$

If the players follow the pointer following algorithm, they only lose if $X > \lceil c_1 n \rceil$, and so their winning probability is at least p_1.

For $k = 1$ the proposition follows from the observation that each player may assign label $n + 1$ to the empty locker. This reduces the problem to an an equivalent game with $n + 1$ players and no empty lockers. ■

To understand the behaviour of Algorithm PF-1 for $k > 1$, let us reconsider Example III.1. The players can be partitioned into two groups, let us call them A and B. Group A are the players which find their labels before opening any empty locker. Each player in Group B opens at least one empty locker before they find his label. So in the example we have A={2,5,9} and B={1,3,4,6,7,8}, which is a partition of the players. Consider player 7, whose label is contained in locker $n + k = 11$. He opens all the empty lockers and finds the labels of all players in B before he finds his label. Since no player in B can find any of the labels in A, Player 7 opens the maximum number of lockers. We will show that this is always the case for the player whose label is in locker $n + k$. Then by analyzing the size of B we can bound the number of lockers the players need to open in order to win with high probability. The idea is formalised in the following theorem.

Theorem IV.2. *Consider a locker game with n players and a fixed number k of empty lockers. For every probability $p < 1$ there is a positive constant $c < 1$ such that if the players can open up to $\lceil c(n + k)\rceil$ lockers, then their success probability using Algorithm PF-1 is at least p.*

Proof: (sketch) We proceed by induction on k, the cases $k = 0, 1$ having been covered in Proposition IV.1. Let A be the set of players who find their label before opening any empty locker, and let B be the remaining players, which are those that open at least one empty locker. No player in A can open a locker which contains a label from B and vice versa. If locker $n + k$ is empty, then no-one will open this locker, so the theorem follows by induction. Otherwise let z be the label in locker $n + k$.

We first show that z opens all empty lockers and finds the labels of all players in B. Indeed, by the behaviour of PF-1, in order to find his label in locker $n + k$ player z must have already opened lockers $n+1, n+2, ..., n+k-1$. Consider any other player j in B. Suppose first that j finds his label before opening locker $n + k$. Since j is in B he must open locker $n + 1$. Since z also opens this locker, he will then follow the same steps as j and find j's label. Now suppose that j opens all empty lockers before finding his label. Then he must open box $n + k$, find label z and then open locker z. From there he follows the same steps as z and finds his label. So z finds label j also. Note in this case j opens the same number of lockers as z.

To complete the proof, we will show there exist two constants c_1 and c_2 such that with probability at least p we have $\lfloor c_1 n\rfloor \leq |B| \leq \lceil c_2 n\rceil$. This is done by analyzing the cycle, C_z, followed by player z. Recall that C_z is the pointer following sequence that starts from locker $n+k$, reaches some empty locker, restarts from locker $n+1$, reaches another empty locker, restarts from locker $n+2$, and so on, and finishes when it comes back to locker $n + k$. Notice that this sequence is nothing other than the prefix of a random sequence, consisting of $1, 2, \cdots, n$ and k 0's, which ends at the k-th 0. This means that the probability that the length L of C_z is at most r is equal to the probability that the length-r prefix of the above random sequence includes all k 0's, and then in turn equal to the probability that in a sample of r items from $\{1, 2, ..., n+k\}$ we find all items $\{n+1, n+2, ..., n+k\}$. Therefore the length of this cycle follows the hypergeometric distribution. So

$$Pr(L \leq r) = \frac{\binom{r}{k}}{\binom{n+k}{r}} = \frac{r(r-1)...(r-k+1)}{(n+k)...(n+1)}$$
$$\leq (\frac{r}{n})^k \leq c^k(1 + k/n)^k$$

where $r = c(n + k)$ for any $0 < c < 1$. There is a similar lower bound, and so asymptotically we have

$$P(L \leq r) = c^k + o(1).$$

By choosing c_1 small enough and c_2 large enough, the theorem follows. ■

We now prove a similar result for Algorithm PF-2(t).

Theorem IV.3. *Consider a locker game with n players and a fixed number k of empty lockers. For every probability $p < 1$ there is a positive constant $c < 1$ such that if the players can open up to $\lceil c(n + k)\rceil$ lockers, then their success probability using Algorithm PF-2(t) is at least p.*

Proof: (sketch) Let L_1 be the number of lockers opened in the first phase of Algorithm PF-2(t) (ie. step 1) and L_2 be the number of lockers opened in the second phase (ie. step 3). Both L_1 and L_2 are governed by the hypergeometric distribution. Indeed, we already showed this for L_2 in Theorem IV.2. For L_1, we see that $L_1 \leq r$ if and only if in a sample of r items drawn from $\{1, 2, ..., n+k\}$ we find t items from the set $\{n+1, n+2, ..., n+k\}$. Therefore we can get tight bounds on L_1 and show that with high probability step 1 can be completed by opening only a constant fraction of lockers. The contents of the remaining lockers are uniformly distributed, and so we may apply Theorem IV.2 to get similar bounds for L_2. The details will be given in the full paper. ■

In the next section we give some experimental results. In particular, we see that Algorithm PF-2(1) gives a much better success probability for the players than Algorithm PF-2(0), but they do best with Algorithm PF-1.

V. Experimental Results

In this section we investigate experimentally the behaviour of Algorithm PF-2(t). The code was written in ansi C and used the GNU random number package to place labels in the lockers. Each experiment was repeated 10,000 times, and the winning probabilities shown are the number of wins for the players divided by 10,000.

In the first table we fix the number of empty lockers at $k = 10$ and the fraction of lockers the players may open at $c = 0.9$. We vary the number $n + k$ of lockers. Recall that t is the number of empty lockers remaining after the first step of the algorithm has completed.

We remark that there is an almost constant winning percentage except for the cases of $t = 0, 1$. Using $t = 1$ the players win with more than three the probability they achieve by using

TABLE I
ALGORITHM PF-2(T): PLAYER WINNING PROBABILITIES WITH $k = 10$ AND $c = 0.9$

$n+k$	t =0	1	2	3	4	5	6	7	8	9	10
100	.06	.23	.31	.34	.34	.33	.33	.33	.34	.34	.34
200	.07	.24	.32	.34	.34	.34	.34	.34	.34	.34	.34
400	.07	.23	.31	.35	.34	.34	.34	.35	.34	.33	.33
800	.07	.23	.31	.34	.34	.35	.35	.35	.33	.35	.35
1600	.07	.24	.32	.34	.35	.34	.35	.34	.35	.34	.34
3200	.07	.24	.32	.35	.35	.36	.35	.35	.35	.35	.35

$t = 0$. The players do best with $t = k$, which is equivalent to Algorithm PF-1. Note that the winning probabilities are almost independent of the number of lockers, $n + k$.

Next we fixed the number of lockers at 2520 (to avoid rounding up) and let the number k of empty lockers vary. The analysis of the previous section involving the hypergeometric distribution suggests that if the players can open about $k/(k + 1)$ lockers they should have a reasonable winning percentage. The results are shown in Table II. Again we observe a remarkably constant winning probability for the players except when $t = 0, 1$. This probability is very close to the empty locker winning probability when the players can open half of the lockers.

TABLE II
ALGORITHM PF-2(T): PLAYER WINNING PROBABILITIES WITH $n + k = 2520$ LOCKERS, $c = k/(k + 1)$ FOR VARIOUS VALUES OF k

k	c	t =0	1	2	3	4	5	6	7	8
1	1/2	.05	.32							
2	2/3	.07	.31	.43						
3	3/4	.08	.30	.40	.42					
4	4/5	.08	.29	.39	.41	.42				
5	5/6	.09	.29	.38	.40	.41	.41			
6	6/7	.08	.29	.37	.39	.40	.39	.40		
7	7/8	.09	.28	.36	.38	.39	.39	.40	.40	
8	8/9	.09	.29	.37	.39	.39	.40	.39	.39	.39

Finally we fixed the number of lockers at 2520, the number of empty lockers at $k = 6$, and let the fraction of lockers that are allowed to be opened to be variable. The results are shown in Table III. We observe that again, apart from $t = 0, 1$, Algorithm PF-2(t) gives an essentially constant winning probability for the players, which increases steadily with c.

TABLE III
ALGORITHM PF-2(T): PLAYER WINNING PROBABILITIES WITH $n + k = 2520$ LOCKERS, $k = 6$ FOR VARIOUS VALUES OF c

c	t =0	1	2	3	4	5	6
1/2	.00	.00	.00	.01	.01	.01	.01
2/3	.00	.02	.05	.07	.08	.08	.09
3/4	.01	.08	.14	.17	.18	.17	.18
4/5	.03	.15	.22	.26	.27	.25	.27
5/6	.05	.22	.30	.33	.34	.33	.34
6/7	.08	.29	.37	.39	.40	.39	.40
7/8	.11	.34	.42	.44	.46	.45	.46
8/9	.14	.39	.47	.49	.50	.49	.50
9/10	.15	.40	.48	.50	.51	.50	.51
11/12	.22	.51	.58	.59	.58	.60	.60
23/24	.44	.74	.77	.78	.78	.77	.78
35/36	.55	.83	.85	.85	.85	.84	.85
71/72	.72	.91	.92	.92	.92	.92	.92

VI. CONCLUSION

We have presented algorithms for the locker problem where there are a constant number of empty lockers. We have shown that the players have a reasonable winning probability if they can open about $k/(k+1)$ of the lockers. Empirically we have seen that they do best by using the modified pointer following Algorithm PF-1, rather than systematically searching for all of the empty lockers. The methods given here do not allow k to be increasing with n. It remains an open problem whether or not the players can achieve a fixed winning probability by opening only a constant fraction of the lockers in this case.

REFERENCES

[1] D. Avis and A. Broadbent. The quantum locker puzzle. ICQNM09, Cancun, February 2009.
[2] E. Curtin and M. Warshauer. The locker puzzle. *The Mathematical Intelligencer*, 28(1):28–31, 2006.
[3] A. Gál and P. B. Miltersen. The cell probe complexity of succinct data structures. In *Proceedings of the 30th International Colloquium on Automate, Languages and Programming (ICALP)*, pages 332–344, 2003.
[4] A. Gál and P. B. Miltersen. The cell probe complexity of succinct data structures. *Theoretical Computer Science*, 379(3):405–417, 2007.
[5] N. Goyal and M. Saks. A parallel search game. *Random Structures and Algorithms*, 27(2):227–234, 2005.