K-Ary Searching With A Lie

Adam Malinowski

Institute of Informatics
Warsaw University

Abstract. We determine the minimal number of queries sufficient to find an unknown integer x between 1 and n if at most one answer may be erroneous. The admissible form of query is: "Which one of the disjoint sets $A_1, \ldots, A_K$ does x belong to?"

1. Introduction

S.M. Ulam in [U] raised the following question: what is the minimal number of yes-no queries sufficient to find an unknown x from set $\{1, \ldots, n\}$ if at most one answer may be a lie. This problem was solved by A. Pelc in [P1] (an analogous problem, for two lies admitted, was solved by W. Guzicki in [G], and for any fixed number of lies by J. Spencer in [S]).

Also A. Pelc in [P2] solved the problem of determining the minimal number of weighings on a beam balance to find counterfeit (heavier) one among n coins if at most one weighing result may be erroneous. It corresponds to the previous problem; the difference is that here the queries admit three possible answers: left pan goes down (which means that x is among the coins on the left pan), right pan goes down (x is on the right pan), the pans are balanced (x is among the remaining coins). There is also the additional condition, due to the physical interpretation, that numbers of coins on left and right pan must be equal.

Our result, solution of Ulam's problem for queries admitting K possible answers, is a generalization of original Ulam's problem (corresponding to the case $K = 2$) as well as the simplified version of the problem of detecting a false coin using beam balance, when we do not require two of the three sets we ask about to have equal sizes (case $K = 3$).

2. Preliminaries

We will follow the notation and terminology of [P1] and [P2].

Let us consider a game played by two players: the Questioner and the Responder. The Responder chooses an integer x from the set $\{1, \ldots, n\}$ and the Questioner has to determine x, asking some queries of form: "Which one of the disjoint sets $A_1, \ldots, A_K$ does x belong to?", where $A_1 \cup \cdots \cup A_K = \{1, \ldots, n\}$ (some of the sets $A_1, \ldots, A_K$ may be empty) and K is a fixed integer not less than 2. The Responder is allowed to lie at most once.

We will show, what is the minimal number of queries, sufficient for the Questioner to find x in the worst case.

At each stage, current state of the game is described by a pair of integers: (a, b). The first number a is the size of L_0, the set of all elements of $\{1, \ldots, n\}$ which satisfy all answers given so far. The second number b is the size of L_1, the set of numbers which satisfy all but one answer. At the beginning of the game the state is $(n, 0)$; the only states, in which x is determined, are $(1,0)$ and $(0,1)$.

The query "Which one of the sets $A_1, \ldots, A_K$ does x belong to?" will be denoted as

$$(x_1, y_1): \ldots :(x_K, y_K),$$

where $x_j = |A_j \cap L_0|$, $y_j = |A_j \cap L_1|$ for $j = \{1, \ldots, K\}$.

The sequence of r identical components of a query $(x_1, y_1): \ldots :(x_1, y_1)$ will sometimes be denoted shorter as $r \cdot (x_1, y_1)$.

It is easy to see, that if the current state of the game is (a, b), the query $(x_1, y_1): \ldots :(x_K, y_K)$, where for each $j \in \{1, \ldots, K\}$ $x_j, y_j \geq 0$, $\Sigma_{j=1}^{K} x_j = a$, $\Sigma_{j=1}^{K} y_j = b$, yields one of the states $(x_1, a - x_1 + y_1), \ldots, (x_K, a - x_K + y_K)$.

Definition 2.1. *i-th weight of a state (a, b):*

$$w_i(a,b) = [(K-1) \cdot i + 1] \cdot a + b \text{ for } i \geq 0.$$

Intuitively, $w_i(a, b)$ is the number of possibilities for the Responder if the current state is (a, b) and the Questioner still has i spare queries.

Lemma 2.2. *With the above notation*

$$\sum_{j=1}^{K} w_{i-1}(x_j, a - x_j + y_j) = w_i(a, b) \text{ for } i > 0.$$

Proof:

$$\begin{aligned}\sum_{j=1}^{K} w_{i-1}(x_j, a - x_j + y_j) &= \sum_{j=1}^{K}\big([(K-1)\cdot(i-1)+1]\cdot x_j + a - x_j + y_j\big)\\ &= [(K-1)\cdot(i-1)+1]\cdot\sum_{j=1}^{K} x_j + \sum_{j=1}^{K} a - \sum_{j=1}^{K} x_j + \sum_{j=1}^{K} y_j\\ &= [(K-1)\cdot(i-1)+1]\cdot a + (K-1)\cdot a + b\\ &= [(K-1)\cdot i + 1]\cdot a + b = w_i(a,b).\end{aligned}$$

∎

Hence, for instance, asking a query such that the weights of all possible answers are equal, reduces the weight of the current state by a factor of K.

Definition 2.3. *Character of a state* (a,b)*:*

$$ch(a,b) = \min\{i: w_i(a,b) \leq K^i\}.$$

Let us denote as $N(a,b)$ the minimal number of queries sufficient to reach $(1,0)$ or $(0,1)$ from the state (a,b). Formally, we define $N(a,b)$ as follows: $N(0,1) = N(1,0) = 0$, and $N(a,b) = i$ if there exists a query $(x_1,y_1):\ldots,:(x_K,y_K)$ for (a,b) such that $N(x_j, a-x_j+y_j) \leq i-1$ for all j, but there is no query such that $N(x_j, a-x_j+y_j) \leq i-2$ for all j.

Lemma 2.4.

(a) *If* $a' \geq a$ *and* $b' \geq b$*, then* $N(a',b') \geq N(a,b)$.

(b) $N(a,b) \geq ch(a,b)$.

Proof:

(a) is obvious.

(b) Let $ch(a,b) = i$. By definition $w_{i-1}(a,b) > K^{i-1}$. Hence by 2.2 any query yields a state (x,y) such that $w_{i-2}(x,y) \geq \frac{1}{K}\cdot w_{i-1}(a,b)$ (the state with the biggest $(i-2)$ weight). Repetition of this argument shows, that any sequence of $i-1$ queries yields a state (x',y') such that $w_0(x',y') = x'+y' > 1$, so the searched number is not found.

∎

We will often use 2.4 without referring to it.

2.4 (b) is the motivation for the following definition:

Definition 2.5. *A state* (a,b) *is nice* $\Leftrightarrow N(a,b) = ch(a,b)$.

3. Which states are nice?

Example 3.1. *If* $K = 2$ *or* 3*, then the state* $(2,0)$ *is nice, because* $ch(2,0) = 3$ *and it is easy to see, that three queries is enough to decide which one of two numbers was chosen. However if* $K \geq 4$*, then* $ch(2,0) = 2$ *and two queries will not suffice to detect the chosen number, because the only sensible query yields the state* $(1,1)$ *and* $ch(1,1) = 2$.

The following lemma shows, that nice states are located quite regularly:

Lemma 3.2. *If* $N(a,b) = i$*, then* $N(a,b+1) = i$ *or* $(a,b+1)$ *is nice.*

Proof: Proceed by induction on i.

If $i = 0$ (hence (a,b) is $(1,0)$ or $(0,1)$), then the thesis is true, because the states $(1,1)$ and $(0,2)$ are nice.

Let now $N(a,b) = i+1$ and $(x_1,y_1):\ldots,:(x_K,y_K)$ be a query for (a,b) such that for all $j \in \{1,\ldots,K\}$ $N(x_j, a-x_j+y_j) \leq i$.

Case 1. $w_{i+1}(a,b) < K^{i+1}$.

Then by 2.2 $\exists j w_i(x_j, a - x_j + y_j) < K^i$. For the state $(a, b+1)$ the query $(x_1, y_1) : \ldots : (x_j, y_j + 1) : \ldots : (x_K, y_K)$ yields the same states as before, except $(x_j, a - x_j + y_j)$, instead of which we have $(x_j, a - x_j + y_j + 1)$. By inductive assumption $N(x_j, a - x_j + y_j + 1) \leq i$ or $(x_j, a - x_j + y_j + 1)$ is nice. But $w_i(x_j, a - x_j + y_j + 1) \leq K^i$, so $ch(x_j, a - x_j + y_j + 1) \leq i$. Hence $N(a, b+1) = i + 1$.

Case 2. $w_{i+1}(a,b) = K^{i+1}$.

First we state an auxiliary lemma:

Lemma 3.3. *If* $w_i(a,b) = K^i, (a,b) \neq (1,0)$*, then* $ch(a, b+1) = i + 1$.

Proof: Immediate by 2.3. ∎

From 2.2 it follows, that $\forall j \; w_i(x_j, a - x_j + y_j) = K^i$. For $(a, b+1)$ the query $(x_1, y_1) : (x_2, y_2) : \ldots : (x_K, y_K)$ yields the states $(x_1, a - x_1 + y_1 + 1)$ and $(x_j, a - x_j + y_j)$ for $2 \leq j \leq K$. By inductive assumption $(x_1, a - x_1 + y_1 + 1)$ is nice (it cannot be $N(x_1, a - x_1 + y_1 + 1) \leq i$, because $ch(x_1, a - x_1 + y_1 + 1) = i + 1$ by 3.3). Hence $N(a, b+1) \leq i + 2 = ch(a, b+1)$, so $(a, b+1)$ is nice. ∎

Corollary 3.4. *If a state* (a,b) *is nice and* $b' > b$*, then the state* (a, b') *is nice.* ∎

The following example illustrates lemma 3.2:

Example 3.5. *Let* $K = 2$. $ch(3,0) = 4$*, but* $N(3,0) > 4$*, because the only sensible query* $(1,0) : (2,0)$ *yields the state* $(2,1)$*, and* $W_3(2,1) = 9 > 2^3$. *Also* $ch(3,1) = 4$*, so* $(3,1)$ *is not nice, hence by 3.2 it requires the same number of queries as* $(3,0)$.

It is easy to see, that $N(3,14) = 5$ ($(1,9) : (2,5)$ is a good first query), and $ch(3,15) = 6$, so $(3,15)$ requires more queries than $(3,14)$, hence by 3.2 $(3,15)$ is nice.

The case of states having form (K^i, b) is the simplest:

Lemma 3.6. *For each* $i \geq 0$ *and* $b \geq 0$ *the state* (K^i, b) *is nice.*

Proof: By 3.4 it is sufficient to show, that $\forall i \; (K^i, 0)$ is nice.

Proceed by induction on i.

For $i = 0$ the thesis is true, because the state $(1,0)$ is nice.

For $(K^{i+1}, 0)$ the query $(K^i, 0) : \ldots : (K^i, 0)$ yields K identical states $(K^i, K^{i+1} - K^i)$, which are nice by inductive assumption and by 2.2 their character is less than the character of $(K^{i+1}, 0)$. ∎

4. Typical states

Definition 4.1. *A state* (a, b) *is typical* $\Leftrightarrow b \geq (K-1)\cdot(a-1)$.

For the initial state $(n, 0)$, where $n = K\cdot d + r, 0 \leq r \leq K-1$, it is clear that the best query is one that splits the state uniformly, that is $r\cdot(d+1, 0) : (K-r)\cdot(d, 0)$. The states we get then, $(d+1, (K-1)\cdot d + r - 1)$ and $(d, (K-1)\cdot d + r)$, are typical.

For fixed a let us introduce the following notation:

$$t(a) = (a, (K-1)\cdot(a-1)),$$

the minimal typical state, and

$$c(a) = \min\{K^i : K^i \geq a\}.$$

In order to calculate $N(n, 0)$ in [P1] and [P2] it was enough to show, that for $K = 2$ and $K = 3$ all typical states are nice. However, as the following example shows, it is not true for bigger K:

Example 4.2. *Let* $K = 4$. *The state* $(5, 12)$ *is typical,* $ch(5, 12) = 3$. *But any query yields a state requiring at least as many queries as the state* $(2, 3)$, *that is more than two (see 3.1).*

We will now determine $N(t(a))$ for $a \leq K^{K-2}$. By 3.2 we will then be able to determine $N(a, b)$ for any typical state (a, b), where $a \leq K^{K-2}$.

Lemma 4.3.

(a) $ch(t(K^i)) = i + 2$ *for* $1 \leq i \leq K-2$,
(b) $ch(t(K^{K-2} + 1)) > K$.

Proof:

(a) $w_{i+2}(t(K^i)) = [(K-1)\cdot(i+2)+1]\cdot K^i + (K-1)\cdot(K^i - 1) \leq [(K-1)\cdot K + 1]\cdot K^i + (K-1)\cdot(K^i-1) = K^{i+2} - K + 1 \leq K^{i+2}$.
$w_{i+1}(t(K^i)) = [(K-1)\cdot(i+1)+1]\cdot K^i + (K-1)\cdot(K^i-1) > K^{i+1}$.
(b) $w_K(t(K^{K-2}+1)) = [(K-1)\cdot K + 1]\cdot(K^{K-2}+1) + (K-1)\cdot K^{K-2} = K^K + K^2 - K + 1 > K^K$.

∎

Now we can prove the following theorem:

Theorem 4.4. *If* $2 \leq a \leq K^{K-2}$, *then* $N(t(a)) = ch(t(c(a)))$.

Proof: If a is equal K^i, then $a = c(a)$ and by 3.6 $t(a)$ is nice, so $N(t(a)) = ch(t(c(a)))$.

It is sufficient to show, that for $0 \leq i \leq K-3$ holds $N(t(K^i + 1)) \geq i + 3$, $(c(K^i + 1) = K^{i+1}$, by 4.3 $ch(t(K^{i+1})) = i + 3$, for $K^i + 1 < a < K^{i+1}$ we use 2.4 (a)). We will show even more, that already $N(K^i+1,0) \geq i+3$ for $0 \leq i \leq K-3$.

It is easy to see, that $N(K^0+1,0) = N(2,0) = 3$ (see 3.1).

Now let $1 \leq i \leq K-3$ and $(x_1,0):\ldots:(x_K,0)$ be the best query for $(K^i+1,0)$. $\Sigma_{j=1}^K x_j = K^i+1$, so $x_j > K^{i-1}$ for some j. By inductive assumption $N(x_j, K^i+1-x_j) \geq (i-1)+3$, hence $N(K^i+1,0) \geq i+3$. ∎

In the next part of this section we will prove, that for $a > K^{K-2}$ the state $t(a)$ (and hence also any typical state) is nice.

Definition 4.5. *Let* $ch(a,b) = i > 0$. *A query* $(x_1,y_1):\ldots:(x_K,y_K)$ *is a splitting* $\Leftrightarrow$

(1) $\forall\, 1 \leq j,j' \leq K, |x_j - x_{j'}| \leq 1$,

(2) $\forall\, 1 \leq j,j' \leq K, |w_{i-1}(x_j, a-x_j+y_j) - w_{i-1}(x_{j'}, a-x_{j'}+y_{j'})| \leq 1$.

Condition (1) in 4.5 assures that a splitting applied to a typical state (a,b) leads to typical states; (2) together with 2.2 assure that the character of the received state is less than $ch(a,b)$. Thus, if there exists a splitting of (a,b) into nice states, then (a,b) is also nice.

Remark 4.6. *If* $K|a$, *then there exists a splitting for state* (a,b).

Proof: Let $a = K\cdot c$, $b = K\cdot d + r$, where $0 \leq r \leq K-1$.

The query $r\cdot(c,d+1):(K-r)\cdot(c,d)$ is a splitting. ∎

The case when $K \not| a$ is more complicated:

Lemma 4.7. *Let* $ch(t(a)) = i+1$, $a = K\cdot c + r$, $1 \leq r \leq K-1$. *If* $i \leq \frac{a-1}{K-r}$, *then there exists a splitting for* $t(a)$.

Proof: Let $(K-1)\cdot(a-1)-(K-1)\cdot(K-r)\cdot i = K\cdot d + r'$, $0 \leq r' \leq K-1$, $d \geq 0$. If $r+r' \leq K$, then

$$r\cdot(c+1,d):r'\cdot(c,d+(K-1)\cdot i+1):(K-r-r')\cdot(c,d+(K-1)\cdot i),$$

and if $r+r' \geq K$, then

$$(K-r)\cdot(c,d+(K-1)\cdot i+1):(K-r')\cdot(c+1,d):(r+r'-K)\cdot(c+1,d+1)$$

is a splitting. ∎

Lemma 4.8. *If* $K \geq 4$, $a > K^{K-2}$, *then there exists a splitting for* $t(a)$.

Proof: We can assume that $K \not| a$ (see 4.6).

By 4.7 it is sufficient to show that if $ch(t(a)) = i+1$, then $i \leq \frac{a-1}{K-1}$.

Let $s = \lfloor \frac{a-1}{K-1} \rfloor$ (the integer part). We will show that $i \leq s$, that is $w_{s+1}(t(a)) \leq K^{s+1}$.

$$a \leq (K-1)\cdot(s+1) \Rightarrow w_{s+1}(t(a))$$
$$\leq [(K-1)\cdot(s+1)+1]\cdot(K-1)\cdot(s+1)+(K-1)\cdot[(K-1)\cdot(s+1)-1]$$
$$= (s+1)^2\cdot(K-1)^2+(s+1)\cdot K\cdot(K-1)-K+1 \leq K^{s+1} \text{ for } s, K \geq 4,$$

but if $a > K^{K-2}$, then $s = \lfloor \frac{a-1}{K-1} \rfloor \geq \frac{K^{K-2}}{K} = K^{K-3} \geq K \geq 4$. ∎

Lemma 4.9.

(a) *For* $K = 2$ *there exists a splitting for* $t(a)$ *if* $a \geq 6$, *and states* $t(a)$ *for* $a < 6$ *are nice.*

(b) *For* $K = 3$ *there exists a splitting for* $t(a)$ *if* $a \geq 9$, *and states* $t(a)$ *for* $a < 9$ *are nice.*

Proof:

(a) Following notation of 4.8 we have $w_{s+1}(t(a)) \leq s^2 + 4\cdot s + 2 \leq 2^{s+1}$ for $s \geq 5$ that is $a \geq 6$.

$t(2)$ and $t(4)$ are nice by 3.6. For $t(3)$ the query $(2,0):(1,2)$ and for $t(5)$ the query $(3,0):(2,4)$ yields nice states (by 3.4) with lower character.

(b) $w_{s+1}(t(a)) \leq 4\cdot s^2 + 14\cdot s + 4 \leq 3^{s+1}$ for $s \geq 4$ that is $a \geq 9$.

$t(3)$ is nice by 3.6. There exists a splitting for $t(6)$ by 4.6, for $t(5)$ and $t(8)$ by 4.7. For $t(2), t(4)$ and $t(7)$ the queries $(1,0):(1,0):(0,2), (2,0):(2,0):(0,6)$ and $(3,2):(3,2):(2,8)$ respectively yield nice states with lower character. ∎

Theorem 4.10. *If* $a > K^{K-2}$, *then the state* $t(a)$ *is nice.*

Proof: If $K = 2$ or $K = 3$, then the thesis follows by 4.9.

Let $K \geq 4$, so by 4.8 there exists a splitting for $t(a)$. For $K^{K-2} < a \leq K^{K-1}$ by 3.2 we have that even if the splitting for $t(a)$ does not lead to a nice state, then it leads to a state requiring exactly K (by 4.3 (a)) queries, and by 4.3 (b) we have $ch(t(a)) > K$, so the state $t(a)$ is nice.

For $a > K^{K-1}$ by induction on $\log_K a$ it follows that the splitting for $t(a)$ leads to a nice state. ∎

5. The main result

In this section we formulate the main result of our paper.

Theorem 5.1. *Let* $n = K\cdot a + r, a \geq 0, 0 \leq r \leq K-1$.

Let us denote

$$p(n) = \begin{cases} (a, (K-1)\cdot a) & \text{if } r = 0, \\ (a+1, (K-1)\cdot a + r - 1) & \text{if } r \neq 0. \end{cases}$$

Then

$$N(n,0) = 1 + \begin{cases} ch(p(n)) & \mathit{if}\, a \geq K^{K-2} \\ \max(ch(p(n)), ch(t(c(a)))) & \mathit{if}\, a < K^{K-2} \text{ and } r = 0, \\ \max(ch(p(n)), ch(t(c(a+1)))) & \mathit{if}\, a < K^{K-2} \text{ and } r \neq 0. \end{cases}$$

Proof: $p(n)$ is the state we achieve after asking the best query for the state $(n, 0)$. If $a \geq K^{K-2}$, then by 4.10 (or 3.6 for $a = K^{K-2}$) $p(n)$ is nice, so it requires $ch(p(n))$ queries. If $a < K^{K-2}$, then the thesis follows by 4.4. ∎

Our proof also shows an algorithm of asking queries. The first one is a query most uniformly splitting the state $(n, 0)$, that is $r \cdot (a+1, 0) : (K - r) \cdot (a, 0)$ and yields a typical state. If current state satisfies the assumptions of 4.6 or 4.7, then we ask a splitting, if it does not, we extend it to the state $c(a)$, adding some elements of $\{1, \ldots, n\}$ excluded by earlier answers. What queries we should ask for such states follows from the proofs of 3.6 and 3.2. Several states, which should be treated individually, are mentioned in 4.9.

References

[G] W. Guzicki, *Ulam's searching game with two lies*, Journal of Combinatorial Theory, Series A **54**, **No.1** (1990), 1–19.

[P1] A. Pelc, *Solution of Ulam's problem on searching with a lie*, Journal of Combinatorial Theory, Series A **44**, **No.1** (1987), 129–140.

[P2] A. Pelc, *Detecting a Counterfeit Coin with Unreliable Weighings*, Ars Combinatoria 27 (1989), 181–192.

[S] J. Spencer, *A Fixed Number of Lies*, to appear.

[U] S.M. Ulam, "Adventures of a mathematician", Scribner, New York, 1976.