Search when the lie depends on the target

Gyula O.H. Katona*
Rényi Institute
Budapest, Hungary
ohkatona@renyi.hu

Krisztián Tichler†
Eötvös University
Budapest, Hungary
ktichler@inf.elte.hu

1 Introduction

Let us start with the basic model of Search Theory. An n-element set S is given, one of its elements, say x, is distinguished, the goal is to find x. Questions of type "$x \in A$?" can be asked, where A is a subset of S. The unknown x should be determined on the base of the answers to these questions. In general one cannot use every subset A. A family $\mathcal{A} \subset 2^S$ is given, the question sets A can be chosen only from $\mathcal{A}$.

We show some "practical examples".

1.1. Twenty question. Alfred chooses a person x, Paul has to find out who he/she is and he can ask questions like "is x a man?", "is x alive?", and so on... Alfred answers honestly, and Paul has to determine the person based on the answers to his questions. Obviously, S is the set of persons, A is the set of men, or the set of living persons, etc.

1.2. Chemical analysis. It is known that the given solution contains exactly one metal. This should be determined by chemical tests. Then S is the set of all metals, $x \in S$ is the one contained in the solution. A chemical test is e.g. when a certain other specific chemical is added to the solution. If $x \in A$ where A is a subset of S then the solution turns red, otherwise it does not.

*The first author was supported by the Hungarian National Foundation for Scientific Research, grant number NK78439.

†The work of the second author was supported by the European Union and co-financed by the European Social Fund (ELTE TÁMOP-4.2.2/B-10/1-2010-0030).

A recent important variant of this example when the solution contains an unknown genetic sequence.

1.3. Criminal investigation. Given a crime, we have a set S of possible perpetrators. The real perpetrator, $x \in S$ should be found. Each evidence restricts x to be in a set $A \subset S$. For instance if a witness says that the perpetrator is bold then we know $x \in A$, where A is the set of bold ones among the possible perpetrators.

There are two basic ways to use the questions. In the *adaptive model* the choice of the next question may depend on the answers to the previous questions. The *search algorithm* starts with a question set A. If the answer is that $x \notin A$ then the next question is a certain A_0, otherwise A_1. If the answer to the question "$x \in A_0$?" is *no* then the question set A_{00} comes, and so on. That is the search algorithm consists of a binary tree structure of subsets of S where A is the root. The information obtained along the path from the root to a leave uniquely determines x. The complexity of such a search algorithm is the length of the longest path from the root to a leaf. The mathematical problem is to find the search algorithm with the least complexity using sets from $\mathcal{A}$. (Shortest algorithm in the worst case.)

In the *non-adaptive model* the question sets are given in advance: $A_1, A_2, \ldots, A_m$. Of course the knowledge if $x \in A_i (1 \leq i \leq m)$ must uniquely determine x. One can easily see that this holds iff $A_1, A_2, \ldots, A_m$ is a *separating family* that is, for any $x, y \in S, x \neq y$ there is an i such that exactly one of $x \in A_i$ and $y \in A_i$ holds. The mathematical problem is to find the minimum of m that is the size of the smallest separating subfamily of $\mathcal{A}$.

There is a very large number of variants of this basic model. The interested reader can find them in the survey paper [11] and in the monographs [10], [3], [2].

An important direction is when the answers to the questions "$x \in A$?" can be wrong. The first such problem was independently posed by Rényi and Ulam. Every subset can be chosen as a question set that is $\mathcal{A} = 2^S$, the search is adaptive, at most one of the answers can be wrong along a path leading from the root to a leave. The unknown x has to be found surely, with probability one. What is the minimum number of questions in the worst case? The problem is called the Rényi-Ulam game ([17], [16]). Berlekamp ([5], [6]) has basically solved the problem (gave good estimates). This problem has many variants, as well. A general term for these types of problems, when the answer to the question can be erroneous *Search with Lies*. [9] is a good survey paper. The case when $\mathcal{A}$ consists of all sets of size at most k is treated

in [12].

2 Our model

In the models Search with Lies briefly introduced in the first section every question has the same chance to be incorrectly answered. In other words, the occurrence of a lie does not depend on the relationship of the question set A and the unknown element x. In our present model this is not true. For a given question there are certain unknowns x triggering the possibility of a false answer. If the unknown x is different from these then the answer must be correct. Let us show some examples continuing our examples in the previous section.

2.1. Twenty question. Suppose that Alfred has a famous transsexual in his mind as the unknown x. Paul asks "is he a man?". Alfred has to answer "yes" or "no". He will unintentionally lie, misleading Paul. (When Twenty Question was played on the Hungarian TV in the nineteen seventies, they had to introduce the the third possible answer "not characteristic" because of the protests concerning incorrect answers of this type.)

2.2. Chemical analysis. The outcome of the chemical test might sensitively depend on a parameter we cannot well control or sense. But only in the case of certain metals. For "good" metals the result of the test is correct, for the "bad" metals however it might be wrong.

2.3. Criminal investigation. The officer asks the witness if the perpetrator is bold. The witness might lie only if it is in his/her interest: the perpetrator is his/her relative or friend.

In the first section a question A divided S into two parts: into A and $\overline{A}$. If the answer was "yes" we learned that $x \in A$, if it was "no" then the conclusion was $x \in \overline{A}$. Here a question is a partition of S into three classes: (A, L, B). If $x \in A$ then the answer is "yes" (or 1), if $x \in B$ then the answer is "no" (or 0), finally if $x \in L$ then the answer can be either "yes" or "no". In other words, if the answer "yes" is obtained then we know that $x \in A \cup L$ while in the case of "no" answer the conclusion is $x \in B \cup L$.

The obvious problem is what the fastest algorithm using such questions is. If there is no limitation on the choice of these 3-partitions, then the easy answer is that only partitions with $L = \emptyset$ should be used and we are back to the old, trivial model. Therefore a natural assumption is that every L is large, that is, $|L| \geq k$ holds for every partition we can use. On the other

hand it will be supposed that all the possible partitions (A, L, B) satisfying $|L| \geq k$ can be used as questions.

The adaptive case will be solved in Section 3 by exhibiting the best algorithm and proving that there is no better one in the worst case. The non-adaptive case is more difficult. In Section 4 we reduce the problem to a graph theoretical problem: a nearly perfect matching in the graph of the n-dimensional cube should be found which satisfies the additional condition that the number of edges in the matching is the same in all directions.

The results of this paper were first presented at the "Workshop on Combinatorial search" in Budapest in April 26th, 2005. Professor Rudolf Ahlswede liked them very much. Each time when we met he urged us to write them up but we kept postponing it. In the mean time he even solved some closely related problems in [1]. I hope he will like that this paper is published at least in his memorial volume.

3 The adaptive search

Suppose that $k \leq n-2$ holds. We start with the description of an algorithm. The starting question is an arbitrary partition (A, L, B) satisfying $|L| = k, |A| = \lceil \frac{n-k}{2} \rceil, |B| = \lfloor \frac{n-k}{2} \rfloor$. After obtaining the answer the unknown x will be restricted either to $A \cup L$ or to $B \cup L$ where $|A \cup L| = \lceil \frac{n+k}{2} \rceil$, $|B \cup L| = \lfloor \frac{n+k}{2} \rfloor$, both sizes are $< n$.

Suppose that x is already limited to a set $Z \subset S$ at a certain stage of the search. The next step of the algorithm will be determined distinguishing two cases depending on the size of Z. However in both cases the new L is chosen to minimize $|Z \cap L|$ since the incorrect answer in L is not interesting outside of Z.

1. $|Z| > n-k$. Choose L of size k in the following way: $S - Z \subset L$. Divide $Z - L$ into two parts A and B of sizes $\left\lceil \frac{|Z-L|}{2} \right\rceil$ and $\left\lfloor \frac{|Z-L|}{2} \right\rfloor$, respectively. This defines the next question (A, L, B).

2. $|Z| \leq n - k$. Choose an L of size k to be disjoint to Z. Divide Z into two parts U and V of sizes $\left\lceil \frac{|Z|}{2} \right\rceil$ and $\left\lfloor \frac{|Z|}{2} \right\rfloor$, respectively. Let the next question (A, L, B) in the algorithm be defined by $A = U, B = V \cup (S - Z - L)$.

After receiving the answer to this last question the unknown element x is restricted to a set Z' of size either $\lceil \frac{n-k}{2} \rceil$ or $\lfloor \frac{n-k}{2} \rfloor$ in the first case and of size either $\lceil \frac{|Z|}{2} \rceil$ or $\lfloor \frac{|Z|}{2} \rfloor$ in the second case. (Observe that all these four values

are less than $|Z|$.)

The algorithm stops when $|Z|$ becomes 1.

Theorem 3.1 *Let $k \leq n-2$. The algorithm described above is the fastest adaptive search.*

Proof. A stronger statement will be proved, namely that this algorithm is the fastest if it is started from a position when the unknown element is restricted to a z-element subset Z. Let $f(n,k,z)$ denote the minimum number of questions in this situation in the worst case. Induction on z will be used.

Suppose that the unknown element is restricted to to a set Z where $|Z| = z$. We will prove that our algorithm is the shortest one, using the assumption that it is the shortest for smaller values of z. Let (A, Y, B) with $|Y| \geq k$ be the first question of an arbitrary algorithm. If the answer is "yes" then the unknown element is restricted to the set $Z \cap (A \cup Y)$, otherwise to $Z \cap (B \cup Y)$. By the inductional hypothesis at least

$$\max\{f(n,k,|Z \cap (A \cup Y)|), f(n,k,|Z \cap (B \cup Y)|)\} \tag{3.1}$$

more questions are needed.

Since $A \cup Y$ and $B \cup Y$ cover Z,

$$\max\{|Z \cap (A \cup Y)|, |Z \cap (B \cup Y)|\} \geq \left\lceil \frac{|Z|}{2} \right\rceil. \tag{3.2}$$

On the other hand either $|A|$ or $|B|$ is at most $\left\lfloor \frac{n-|Y|}{2} \right\rfloor \leq \left\lfloor \frac{n-k}{2} \right\rfloor$. Hence the smaller one of $|Z \cap A|$ and $|Z \cap B|$ is also at most $\left\lfloor \frac{n-k}{2} \right\rfloor$. This implies

$$\max\{|Z \cap (A \cup Y)|, |Z \cap (B \cup Y)|\} \geq |Z| - \left\lfloor \frac{n-k}{2} \right\rfloor. \tag{3.3}$$

Using the obvious fact that $f(n,k,z)$ is a monotone function of z, (3.1)-(3.3) imply

$$f(n,k,z) \geq 1 + \max\{f(n,k,|Z \cap (A \cup Y)|), f(n,k,|Z \cap (B \cup Y)|)\} \geq$$

$$1 + \max\left\{ f\left(n,k,\left\lceil \frac{z}{2} \right\rceil\right), f\left(n,k,z - \left\lfloor \frac{n-k}{2} \right\rfloor\right)\right\}.$$

Here $\left\lceil \frac{z}{2} \right\rceil \geq z - \left\lfloor \frac{n-k}{2} \right\rfloor$ holds if and only if $z \leq n-k$, following the separation in the definition of the algorithm proving that we cannot do anything better than our algorithm. □

One can conclude that the best algorithm decreases the size of Z by $\lfloor \frac{n-k}{2} \rfloor$ in each step until its size becomes at most $n-k$. Then the usual "halving" finishes the algorithm. Using the trivial fact $f(n,k,1) = 0$, this gives us a formula for the length of the algorithm.

Consequence 1 *Suppose $k \leq n-2$. The length of the fastest adaptive algorithm is*

$$f(n,k,n) = f(n,k) = \left\lceil \frac{n}{\lfloor \frac{n-k}{2} \rfloor} \right\rceil - 2 + \left\lceil \log_2 \left(n - \left\lfloor \frac{n-k}{2} \right\rfloor \left(\left\lceil \frac{n}{\lfloor \frac{n-k}{2} \rfloor} \right\rceil - 2 \right) \right) \right\rceil .$$

It is worth mentioning that this formula is basically identical with that of Theorem 3.8 in [11].

$k \leq n-2$ was supposed in Consequence 1. If $k = n$, the tests give no information, the unknown element cannot be found. The case $k = n-1$ is not really better. Let the question contain L as an arbitrary $n-1$-element set , the remaining one-element set is A. If the answer is "yes" then we obtained no information. On the other hand, if the one-element set is B then the answer "no" leaves us without information. That is in the worst case no information is gained from these questions.

4 The non-adaptive search

In this case the "algorithm" consists of a series of questions

$$(A_1, L_1, B_1), (A_2, L_2, B_2), \ldots, (A_m, L_m, B_m) \tag{4.1}$$

such that the answers to these questions uniquely determine x in all cases. Take two distinct elements $x, y \in S$. If

$$\text{either} \quad x \in A_i, y \in B_i \quad \text{or} \quad x \in B_i, y \in A_i \tag{4.2}$$

holds for the question (A_i, L_i, B_i) we say that this question *really separates* x and y. If (4.2) holds then the answer to this question will be different when x is the unknown element and when it is y. In other words this question

distinguishes x and y. On the other hand, if both x and y are in A_i (B_i) then the answer to the question is the same in the two cases (when x is the unknown or it is y). Finally, if one or both x and y are in L_i then we might obtain the same answer in the two cases, this question does not necessarily distinguishes x and y.

One can see from this that the answers to the set of questions (4.1) uniquely determine the unknown x iff (4.2) holds for every pair $x, y \in S$. We say in this case that (4.1) is a *really separating* set of questions. Our goal is to minimize m under the conditions that (4.1) is really separating and $|L_i| \geq k$, for given n, k. Let this minimum be denoted by $N(n, k)$.

It is useful to consider the "characteristic matrix" of the set of questions. The characteristic vector associated with the question (A, L, B) is a vector containing 1, $*$, and 0 in the jth coordinate if the jth element of S is in A, L, B, respectively. Let the $m \times n$ question-matrix Q have the characteristic vector associated with (A_i, L_i, B_i) in its ith row. Condition (4.2) is equivalent to the condition that for any pair of distinct columns of Q there is a row where the entries are 0, 1 or 1, 0 in the crossing points of this row and the two given columns. We say that that such a matrix is $*$*-less separating*. In these terms $N(n, k)$ is the minimum number of rows in an $m \times n$, $*$-less separating 0,$*$,1-matrix containing at least k stars in each row.

The following trivial lemma will be used later.

Lemma 4.1 *$2^x \geq 2x$ holds for every non-negative integer x.*

Proof. The statement is true for $x = 0, 1, 2$. For $x \geq 3$ one can use induction: $2^x = 2^{x-1} + 2^{x-1} \geq 2(x-1) + 2 = 2x$. □

Lemma 4.2 *If Q is an $m \times n$, $*$-less separating 0,$*$,1-matrix containing at least k stars in each row then*

$$2km \leq 2^m \tag{4.3}$$

holds.

Proof. Let m_j denote the number of $*$s in the jth column of Q. Replacing all $*$s in the jth column by either 0 or 1, 2^{m_j} different columns are obtained. Consider another, say the ℓth column. Since Q is $*$-less separating, the

columns obtained from the ℓth column by replacing the $*$s by 0 or 1 must be different from the columns obtained from the jth column. Hence we have

$$\sum_{j=1}^{n} 2^{m_j} \leq 2^m. \tag{4.4}$$

Lemma 4.1 gives a lower estimate on the left hand side:

$$\sum_{j=1}^{n} 2^{m_j} \geq \sum_{j=1}^{n} 2m_j = 2\sum_{j=1}^{n} m_j. \tag{4.5}$$

The last sum in (4.5) is just the total number of $*$s in Q therefore it must be at least km (at least k in each of the m rows).

$$\sum_{j=1}^{n} m_j \geq km. \tag{4.6}$$

Inequalities (4.4)-(4.6) give (4.3). □

Lemma 4.3 *If Q is an $m \times n$, $*$-less separating 0,$*$,1-matrix containing at least k stars in each row then*

$$n + km \leq 2^m \tag{4.7}$$

holds.

Proof. It will be very similar to the proof of the previous lemma. We use here a tiny bit improved version of Lemma 4.1. When $x = 0$ then $2^0 = 1$ is used rather than $2^0 \geq 2 \cdot 0$. (4.5) becomes

$$\sum_{j=1}^{n} 2^{m_j} \geq \sum_{j=1}^{n} m_j + \sum_{j=1}^{n} m_j + (\text{the number of } j\text{s with } m_j = 0). \tag{4.8}$$

Here

$$\sum_{j=1}^{n} m_j + (\text{the number of } j\text{s with } m_j = 0) \geq n \tag{4.9}$$

since the non-zero m_js are decreased by replacing them by 1. Use (4.6) for the first term of the right hand side of (4.8) then (4.8) for the two other terms:

$$\sum_{j=1}^{n} 2^{m_j} \geq km + n.$$

(4.4) finishes the proof. □

It is somewhat surprising that these two easy conditions (Lemmas 4.2 and 4.3) are sufficient for the existence of a good Q.

Theorem 4.1 *Suppose $3 \leq m$. A Q $m \times n$, $*$-less separating 0,$*$,1-matrix containing at least k stars in each row exists if and only if both (4.3) and (4.7) hold.*

Proof. *Sketching why we need here a graph construction.* We only have to construct a matrix satisfying the conditions if the inequalities (4.3) and (4.7) hold. The matrix will contain one or zero $*$s in every column, and exactly k $*$s in every row. The 0,1 columns of the matrix will be considered as points of the m-dimensional cube B_m. (Here $B_m = (V, E)$ is a graph where V consists of all 0,1 sequences of length m and two such vertices are adjacent if the sequence differ in exactly one position.) A column containing one $*$ can be considered as a pair of points, namely the points corresponding to the two columns obtained by replacing the $*$ by a 0 and a 1. These points are adjacent in B_m therefore the column containing exactly one $*$ can be considered as an edge of B_m. This edge has a *direction*, namely the index of the position of the $*$. It is obvious that two such edges cannot have a common point, otherwise the two columns would not be different by all substitutions. This shows that our matrix generates a matching in B_m. Since we want to have exactly k $*$s in every row, the number of edges in the desired matching should be the same in every direction.

A subgraph (in our case a matching) of B_m is called *balanced* if the number of edges in every direction is the same. We showed how these concepts came into the picture. Let us now formulate our main tool what was developed for the present purpose but its proof can be found in [14].

Theorem 4.2 *$B_m(m \geq 3)$ contains a balanced matching with*

$$\left\lfloor \frac{2^{m-1}}{m} \right\rfloor \tag{4.10}$$

edges in every direction.

The construction. Suppose that (4.3) and (4.7) hold. Start with the balanced matching in Theorem 4.2. By (4.3) k cannot exceed (4.10). Keep only k edges of the matching in each direction. If e is an edge of the matching

in direction i then take a a corresponding column in Q having a $*$ in the ith row, its other 0,1 entries are the joint coordinates of the two endpoints of e. In this way we obtained an $m \times km$ $*$-less separating matrix. We need to add $n - km$ 0,1 columns (without a $*$) keeping the property. The existing km columns exclude $2km$ columns, what are obtained by replacing the $*$s by 0 or 1. There are $2^m - 2km$ other 0,1 columns for our disposal. However $n - km \leq 2^m - 2km$ follows from (4.7), the construction of Q can be completed. □

Consequence 2 *If $k \geq n - 2 \geq 1$ then the minimum length of the non-adaptive algorithm is*

$$N(n,k) = \min\{m : \ 2km \leq 2^m, n + km \leq 2^m\}.$$

The conditions on n and k ensure $m \geq 3$ by (4.7). Theorem 4.2 can be applied. □

5 Remarks

1. Pálvölgyi (unpublished) [15] gave an asymptotically good construction for the non-adaptive case. Bassalygo and Kabatianski (unpublished) [4] also solved a problem related to the non-adaptive case.

2. There were earlier attempts to model the situation described in the paper. Katona and Szemerédi [13] considered the non-adaptive case (formulated in terms of graphs), when the partitions $(A_1, L_1, B_1), (A_2, L_2, B_2), \ldots, (A_m, L_m, B_m)$ really separate every pair of elements x and y. It was proved that

$$\sum_{i=1}^{m} |A_i| + \sum_{i=1}^{m} |B_i| \geq n \log_2 n$$

that is if the cost of a test is the number of "real elements" then one cannot do better than taking $L_i = \emptyset$ for every i and "halve" the underlying set $\log_2 n$ times. For what powers of $|A_i|$ and $|B_i|$ is it still true? For recent improvements see [7] and [8].

3. We have to admit that the condition that all L's have size at least k is not realistic from a practical point of view. In a typical case many L_i's can be empty. However if the partitions with large L_i are numerous and

situated adversely then it can reduce the ideal minimum length $\log_2 n$. Find conditions for that in both the adaptive and non-adaptive cases.

An interesting generalization of our model in the present paper is the following. Let a test be a family $\{A_1, A_2, \ldots, A_t\}$ where their union covers the underlying set (set of possible unknown elements). If the only unknown element is in $A_{i_1} \cap \ldots \cap A_{i_u}$ then the result of the test is any one of the indices $i_1, \ldots, i_u$. One can ask mathematical questions similar to the ones in our paper.

6 Acknowledgements

The first author is indebted to Professor Rudolf Ahlswede for his lifetime friendship and encouragement.

The second author is grateful to Professor Ahlswede for offering him a 6 month pre-doctoral fellowship within Marie-Curie program in 2004/05.

We are also indebted to the anonymous referees for helping to improve the presentation of the results.

References

[1] Rudolf Ahlswede, General theory of information transfer: updated, General Theory of Information Transfer and Combinatorics, Special Issue of *Discrete Applied Mathematics* **156**(9)(2008) 1348-1388. (Online available at: http://www.math.uni-bielefeld.de/ahlswede/homepage/public/220.pdf)

[2] Rudolf Ahlswede and Ingo Wegener, *Search Problems*, Wiley Interscience Series in Discrete Mathematics John Wiley & Sons Inc. 1980.

[3] Martin Aigner, *Combinatorial Search*, John Wiley & Sons, Inc. New York, NY, USA, 1988.

[4] Leonid Bassalygo and Grigori Kabatianski, personal communication.

[5] E.R. Berlekamp, Block coding for the binary symmetric channel with noiseless, delayless feedback, in *H.B. Mann, Error Correcting Codes*, Wiley, 1968.

[6] E.R. Berlekamp, R. Hill and J. Karim, The solution of a problem of Ulam on searching with lies, *IEEE Int. Symp. on Inform. Theory*, MIT, Cambridge, MA USA, 1998, pp. 244.

[7] Béla Bollobás, Alex Scott, On separating systems *European J. Combin.* **28**(4)(2007) 1068-1071.

[8] Béla Bollobás, Alex Scott, Separating systems and oriented graphs of diameter two, *J. Combin. Theory Ser. B* **97**(2)(2007) 193-203.

[9] Christian Deppe, Coding with Feedback and Searching with Lies, in: *Entropy, Search, Complexity* Eds: I. Csiszár, G.O.H. Katona, G. Tardos, Bolyai Society Mathematical Studies **16**(2007), pp. 27-70.

[10] D.-Z. Du and F.K. Hwang *Combinatorial Group Testing*, World Scientific, 1993.

[11] G. Katona, Combinatorial Search Problems, A survey of combinatorial theory (Ed. by J.N. Srivastava, North Holland/American Elsevier, Amsterdam/New York, 1973) 285-308.

[12] Gyula O.H. Katona, Search with small sets in presence of a liar, J. Statictical Planning and Inference 100 (2002) 319-336.

[13] G. Katona and E. Szemerédi, On a problem of graph theory, *Studia Sci. Math. Hungar.* **2**(1967) 23-28.

[14] Gyula O.H. Katona and Krisztián Tichler, Existence of a balanced matching in the hypercube, submitted.

[15] Dömötör Pálvölgyi, personal communication.

[16] A. Rényi, On a problem of information theory, *MTA Mat. Int. Közl.* **6B**(1961) 505-516.

[17] S. Ulam, *Adventures of a Mathematician*, Scribner, New York, 1976.