Searching with Permanently Faulty Tests

Andrzej Pelc

Département d'Informatique
Université du Québec à Hull
C.P. 1250, succursale "B"
Hull, Québec J8X 3X7, Canada

Abstract. We investigate searching strategies for the set $\{1,\dots,n\}$ assuming a fixed bound on the number of erroneous answers and forbidding repetition of questions. This setting models the situation when different processors provide answers to different tests and at most k processors are faulty. We show for what values of k the search is feasible and provide optimal testing strategies if at most one unit is faulty.

Introduction.

The problem of coping with erroneous information in discrete search procedures has recently attracted attention of several authors (cf. [**10, 8, 9, 4, 5, 6, 1, 2**]). It is usually formulated in terms of a two-person game between the Questioner and the Responder. The latter chooses an element $x \in \{1,\dots,n\}$ unknown to the Questioner who has to find it by stating questions of a prescribed form: most often either comparisons, that is, questions of the form $x < a$? for $a \in \{1,\dots,n\}$ or arbitrary yes-no queries, that is, questions of the form $x \in T$? for $T \subset \{1,\dots,n\}$. The Responder gives answers some of which can be erroneous; the number of errors, however, is limited in a given way. The problem is to find an optimal winning strategy of the Questioner.

Two further points have to be specified in order to make the rules of the game complete. The first is the way of limiting errors: their number can be either bounded for the entire game (cf. [**8, 9, 4, 1, 2**]) or may depend on the number of questions; the latter is the case when a bound on the fraction of errors is imposed (cf. [**5**]) or when the Responder lies with a given probability (cf. [**7, 5**]). The second point which should be made precise is the influence of previous answers on subsequent queries: the search can be either carried out interactively (that is the Questioner knows the answer to a query at the moment of stating the next one and is able to modify his behaviour accordingly) or non interactively (all the questions have to be stated at once, then all the answers collected and the unknown number detected).

The above described game models a situation in processing and sending information which can be altered by some kind of "noise". The aim of constructing optimal strategies of the Questioner is to find the most efficient way of recovering original information in spite of this possible deterioration. An information

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada, grant A8136.

channel alters messages with a given (usually small) probability. This setting was studied in [7, 5]. For very small values of this probability it may be reliably assumed that no more than k errors will occur during the entire transmission which justifies interest in the version of the game with a fixed bound on the number of errors. It is well known (cf. for example, [6]) that non-interactive search with arbitrary yes-no queries assuming at most k errors is equivalent to finding k-error correcting codes. Actually, optimal Questioner's strategies in this version of the game yield shortest k-error correcting block codes of a given size. Finding such codes is considered to be the main problem of coding theory (cf. [3]).

In the above model each erroneous answer is counted separately even if the same query was repeated many-times. This is explained by the fact that the actual answers are assumed correct, they are just sometimes altered during transmission, each of them independently. Thus, a simple (though usually far from optimal) questioning strategy is to repeat each question sufficiently many times and to take the majority answer.

Suppose, however, that erroneous answers are not due to noisy transmission but to faulty hardware units responsible for providing them. In this situation, of course, we cannot expect that repeating the same query can help detecting the truth: a processor which gave a wrong answer once may keep repeating the error all the time. It is, thus, reasonable to assume a bound on the number of bad units rather than limiting erroneous answers.

This assumption can also be modelled by the above described searching game between the Questioner and the Responder; however, the rules of stating queries need to be suitably modified. We assume that each test of the type $x \in T$? ($T \subset \{1, \ldots, n\}$) is processed by a unit at most k of which can be faulty. Since information obtained from the tests T and $\{1, \ldots, n\} \setminus T$ is equivalent, we assume that both such tests are processed by the same unit but tests providing non-equivalent information (that is, different and non-complementary) — by different units. This is translated into the rules of our searching game in the following way: the Questioner can ask questions of the type $x \in T$? with all sets T pairwise different and non-complementary. Throughout the paper such queries are called non-repetitive tests. As before, at most k lies are possible among answers. We study both the interactive and non-interactive version of search characterizing when it can be performed and establishing optimal searching strategies in special cases.

It should be noted that if repeating questions is forbidden, limiting them just to comparison form makes the search impossible. Indeed, even with one faulty unit, comparison search in $\{1, \ldots, n\}$ for $n > 1$ cannot be carried out. This justifies allowing arbitrary tests of the form $x \in T$?

Feasibility of search.

As opposed to the situation of questioning with admissible repetitions, when each fixed number of errors can be overcome, a large number of possibly faulty units can make successful searching entirely impossible. The following proposition gives an exact bound on the number of errors which do not jeopardize non-repetitive testing.

Proposition. *It is possible to find an unknown* $x \in \{1, \ldots, n\}$ *by non-repetitive testing assuming* k *errors if and only if* $k < 2^{n-3}$.

Proof: Since we do not care at present about efficiency of search but only its feasibility, there is no difference between the interactive and non-interactive case: we may use a maximal family of non-repetitive tests. Clearly, it is possible to complete the search assuming k errors if and only if each pair of integers x, $y \in \{1, \ldots, n\}$ is separated by at least $2k + 1$ tests of such a family. (A test T is said to separate x and y if $x \in T$ and $y \notin T$ or if $x \notin T$ and $y \in T$. Each such test contributes 1 to the Hamming distance between x and y in the respective binary encoding). Take any maximal family F of non-repetitive tests and fix a pair of distinct integers x and y. There are 2^{n-1} sets separating x and y: each of the form $\{x\} \cup A$ or $\{y\} \cup A$ for all $A \subset \{1, \ldots, n\} \setminus \{x, y\}$. Each $\{x\} \cup A$ has a complement $\{y\} \cup \{1, \ldots, n\} \setminus \{x, y\} \setminus A$ and, consequently, exactly one from each such pair of complements is in F. Therefore, there are 2^{n-2} tests in F separating x and y. This gives the necessary and sufficient condition $2k + 1 \leq 2^{n-2}$ for the feasibility of search with k faulty units, which is equivalent to $k < 2^{n-3}$. This completes the proof. ▮

Interactive searching strategies.

In this section we discuss optimal non-repetitive interactive searching strategies assuming at most one error. In the setting which admitted repetitions the minimal number of questions sufficient to find $x \in \{1, \ldots, n\}$ in the interactive case was obtained by Pelc [4]. Our next result shows that the same efficiency can be achieved with non-repetitive questioning provided that $n > 3$. For $n \leq 3$ non-repetitive search with one error cannot be carried out, as implied by the Proposition.

Theorem 1. *The minimal number of interactive non-repetitive tests sufficient to find* $x \in \{1, \ldots, n\}$ *for* $n > 3$*, assuming at most one error, is equal to:*

$$\min\{k : n(k+1) \leq 2^k\}, \text{ if } n \text{ is even;}$$
$$\min\{k : n(k+1) + (k-1) \leq 2^k\}, \text{ if } n \text{ is odd.}$$

Proof: We first recall some terminology from [4]. With each stage of the game, when the turn of the Questioner comes, we associate a state of the game which is a pair (a, b) of integers. The first of them is the size of the truth set: the set of those elements of $\{1, \ldots, n\}$ which satisfy all answers given previously. The second integer is the size of the lie set: the set of those elements of $\{1, \dots, n\}$ which satisfy all but one answer. We define the weight of a state (a, b) correspoding to a stage of the game at which j questions remain. This is

$$w_j(a, b) = a(j + 1) + b.$$

Any question asked in the state (a, b) yields two states (a_1, b_1) and (a_2, b_2) corresponding to answers "yes" and "no", respectively. The question "Is the unknown number in the subset A of size x of the truth set or in the subset B of size y of the lie set?" is noted as $[x, y]$? and the resulting answers are represented schematically

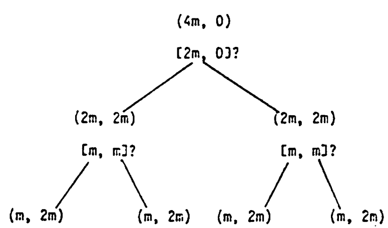
(a, b)
$[x, y]$?
/ \
(a_1, b_1) (a_2, b_2)

with answer "yes" to the left and answer "no" to the right. For every state (a, b) we define its character

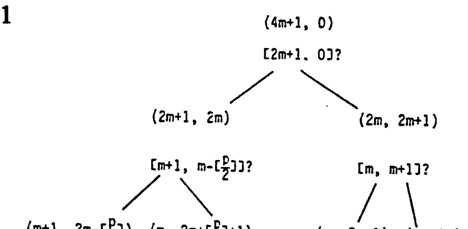
$$ch(a, b) = \min\{k: w_k(a, b) \le 2^k\}.$$

In [4] an optimal (possibly repetitive) questioning strategy was described whose first two steps are the following (depending on divisibility of n by 4):

1. $n = 4m$

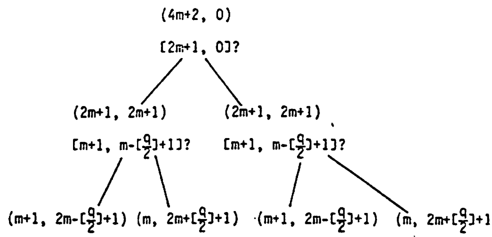


2. $n = 4m + 1$



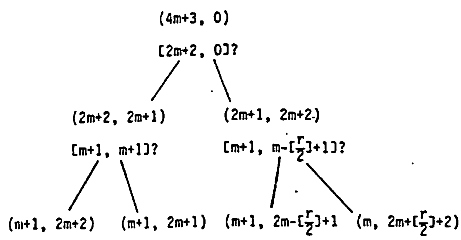
where $p = ch(2m + 1, 2m)$.

3. $n = 4m + 2$



where $q = ch\,(2m + 1, 2m + 1)$.

4. $n = 4m + 3$



where $r = ch\,(2m + 1, 2m + 2)$.

First suppose $n \geq 8$, hence $m \geq 2$. It is easy to see that the above described first two questions are always non-repetitive and, in fact, they are about sets whose symmetric difference has size bigger than 2. However, the optimal strategy constructed in [4] includes repetitions in further stages. We may assume, without loss of generality, that it does not contain questions $x \in T$? and $x \in \{1, \ldots, n\} \setminus T$? because the second can be equivalently changed to a repetition of the first reversing "yes" and "no" answers.

Our aim is to modify the strategy described in [4] into non-repetitive testing by changing some questions without adding extra ones. This is enough to prove the theorem because our formula for the minimal number of non-repetitive tests coincides with that from [4] for questions with possible repetitions.

First observe that in view of optimality no question in the original strategy could be repeated more than 3 times. Indeed, assuming only one lie the true answer is certainly known after 3 repetitions and further repetitions are pointless. Hence, if we show that after the first two questions at least two elements e, f are ruled out, that is, are neither in the truth set nor in the lie set, we may easily transform the original questioning strategy into a non-repetitive one. Indeed, let $T \subset \{1, \ldots, n\} \setminus \{e, f\}$. Since the first two tests have symmetric difference of more than two elements, at most one of them can be among sets $T \cup \{e\}, T \cup \{f\}$, $T \cup \{e, f\}$. All further questions are of the form $T \subset \{1, \ldots, n\} \setminus \{e, f\}$. If such a test T occurs for the second time in the original searching strategy, change it to the first of $T \cup \{e\}$, $T \cup \{f\}$, $T \cup \{e, f\}$ not yet used; if it occurs for the third time change it to the second unused of them. Clearly, the modified questions are equivalent to the original ones because the additional elements were ruled out previously.

In each of the 4 cases described above consider the four states arising after two questions and answers. Call the elements which are neither in the truth set nor in the lie set, free, and consider the number of free elements in each of the four resulting states.

Case 1. $n = 4m$. The number of free elements is m in each of the four states.

Case 2. $n = 4m + 1$. The number of free elements is: $m + \left[\frac{p}{2}\right]$, $m - \left[\frac{p}{2}\right]$, m, $m + 1$, respectively, where $p = ch\,(2m + 1, 2m)$.

Case 3. $n = 4m + 2$. The number of free elements is: $m + \left[\frac{q}{2}\right]$, $m - \left[\frac{q}{2}\right] + 1$, $m + \left[\frac{q}{2}\right]$, $m - \left[\frac{q}{2}\right] + 1$, respectively, where $q = ch\,(2m + 1, 2m + 1)$.

Case 4. $n = 4m + 3$. The number of free elements is: m, $m + 1$, $m + \left[\frac{r}{2}\right] + 1$, $m - \left[\frac{r}{2}\right] + 1$, respectively, where $r = ch\,(2m + 1, 2m + 2)$.

Since by our assumption $m \geq 2$, the number of free elements in Case 1 is always at least 2. In Case 2 it is enough to prove

$$(\star) \quad m - \left[\frac{p}{2}\right] \geq 2,$$

in Case 3 – to prove

$$(\star\star) \quad m - \left[\frac{q}{2}\right] + 1 \geq 2,$$

and in Case 4 – to prove

$$(\star\star\star) \quad m - \left[\frac{r}{2}\right] + 1 \geq 2.$$

$(\star)$ follows from $2m - 2 \geq p$ which is in turn implied by the inequality $w_{2m-2}(2m + 1, 2m) = (2m + 1)(2m - 1) + 2m \leq 2^{2m-2}$, true for $m \geq 5$.

$(\star\star)$ follows from $2m \geq q$ which is in turn implied by the inequality $w_{2m}(2m + 1, 2m + 1) = (2m + 1)(2m + 1) + (2m + 1) \leq 2^{2m}$, true for $m \geq 3$.

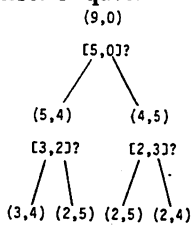
$(\star\star\star)$ follows from $2m \geq r$ which is in turn implied by the inequality $w_{2m}(2m + 1, 2m + 2) = (2m + 1)(2m + 1) + (2m + 2) \leq 2^{2m}$, true for $m \geq 3$.

Hence, our result is proved:

- for $n = 4m$, if $m \geq 2$,
- for $n = 4m + 1$, if $m \geq 5$,
- for $n = 4m + 2$, if $m \geq 3$,
- for $n = 4m + 3$, if $m \geq 3$.

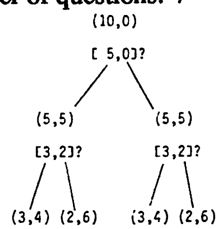
It remains to be proved for $4 \leq n \leq 7$ and for $n = 9, 10, 11, 13, 17$. For $n = 9, 10, 11, 13, 17$ we describe the first two tests carried out interactively in such a way that there are at least two free elements in each of the four resulting states. Moreover, the character of the resulting states is smaller by at most 2 than the minimal number of tests sufficient at the beginning and they are all of the form (a, b) with $b \geq a$. By an argument from [4] it is possible to complete a (possibly repetitive) search in $ch(a, b)$ steps starting from such a state (a, b). Thus, using the free elements as before to modify some further questions if necessary, it is possible to complete the entire search in an optimal and non-repetitive way.

$n = 9$, minimal number of questions: 7



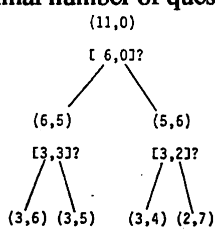
ch(3,4), ch(2,5), ch(2,4) ≤ 5

$n = 10$, minimal number of questions: 7



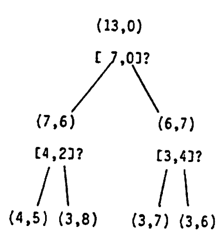
ch(3,4), ch(2,6) ≤ 5

$n = 11$, minimal number of questions: 7



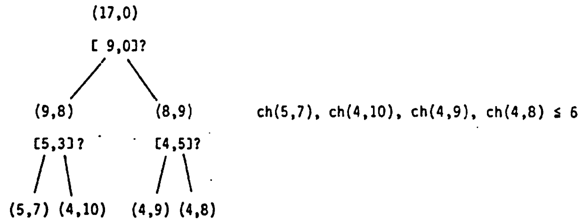
ch(3,6), ch(3,5), ch(3,4), ch(2,7) ≤ 5

$n = 13$, minimal number of questions: 7



ch(4,5), ch(3,8), ch(3,7), ch(3,6) ≤ 5

$n = 17$, minimal number of questions: 8



Hence, our proof is complete for $n \geq 8$.

For $n = 5, 6, 7$, we simply give 6 non-repetitive tests all at once. In view of [4] this cannot be improved even allowing repetitions. Hence, for those values of n we provide at the same time an optimal non-interactive searching strategy.

$n = 5$
tests: $\{1\}, \{2,3\}, \{2,4\}, \{2,5\}, \{3,4\}, \{4,5\}$.
$n = 6$
tests: $\{1,2,3\}, \{1,4,5\}, \{1,3,5\}, \{3,4\}, \{2,5\}, \{1,6\}$.
$n = 7$
tests: $\{4,5,6\}, \{2,3,6\}, \{1,3,5\}, \{3,4,7\}, \{2,5,7\}, \{1,6,7\}$.

It remains to consider the case $n = 4$. In view of [4] the minimal number of questions is now 5. We propose the following interactive non-repetitive strategy. The first two tests are $\{1,2\}, \{1,3\}$. Any of the four sequences of answers results in one element a in the truth set and two elements b, c in the lie set. Three additional tests $\{a\}$, $\{b\}$ and $\{c\}$ complete the search. Thus, for $n = 4$ as well, there is a non-repetitive strategy using the minimal number of 5 questions.

This concludes the proof of the theorem. In fact, not only did we show the length of an optimal strategy, but an optimal algorithm itself was described in our proof. ∎

Non-interactive searching strategies.

In this section we turn attention to non-interactive non-repetitive searching strategies assuming at most one error. We are not able to decide if non-repetitive testing can be as efficient as an optimal questioning strategy with repetitions allowed. The reason is that optimal non-interactive strategies allowing repetitions are unknown in general, even for one error: as mentioned in the introduction they are equivalent to finding shortest 1-error correcting codes.

Nevertheless, we are going to construct optimal non-interactive non-repetitive searching strategies supposing at most one error in the case when the size of the search space is $n = 2^m$. With repetitions allowed optimal strategies are given in this case by Hamming codes (cf. for example, [3]).

Although these actually involve repetitions in general, we will describe a way of constructing such codes which yields non-repetitive testing. In view of Proposition this cannot be done for $m = 1$. The case $m = 2$ will be handled later separately, so we assume $m \geq 3$.

Theorem 2. *Let $n = 2^m$, for $m \geq 3$. The minimal number of non-interactive non-repetitive tests sufficient to find $x \in \{1, \ldots, n\}$, assuming at most one error is $k(m) = \min\{k: k < 2^{k-m}\}$.*

Proof: The lower bound in our theorem is equal to the length of a 1-error correcting Hamming code with m information bits. These codes are shortest possible, hence, it is enough to show how to construct a matrix of such a Hamming code which yields non-repetitive testing. More precisely the resulting code should consist of such words $a_1, \ldots, a_n$ that $a_i = (a_{i1}, \ldots, a_{ik(m)})$, $a_{ij} = 0, 1$, and the family $\{T_j: j \leq k(m)\}$, $T_j = \{a_i: a_{ij} = 1\}$, is composed of distinct non-complementary sets. We will show that this is the case if the $k(m) \times (k(m) - m)$ matrix generating the code satisfies some simple properties. Let this matrix be given in the normal form $M = \left[\frac{B}{I_{k(m)-m}}\right]$ with the identity matrix corresponding to verification bits in the lower part.

Claim:

The resulting code is as required above, provided that B satisfies the following properties:

1. rows are distinct;
2. columns are distinct and non-complementary; ($(c_1, \ldots, c_n)$ and $(c'_1, \ldots, c'_n)$ are complementary if $c'_i = 1 - c_i$ for all i).
3. each row contains at least two digits "1";
4. each column contains at least two digits "1".

(Note that properties 1. and 3. are usual requirements for a matrix generating a Hamming code).

Let the codewords $a_1, \ldots, a_n$, where $a_i = (a_{i1}, \ldots, a_{ik(m)})$ be displayed as rows of a $n \times k(m)$ matrix $[a_{ij}]$. If the above defined sets T_i contained an identical or complementary pair this would mean that two columns, p and q of this matrix are identical or complementary. Since every binary sequence of length m is an initial segment of some codeword (first m bits are information bits), it is impossible that $p, q \leq m$. Suppose that $p \leq m$ and $q > m$ and let $q_0 = q - m$. Take the q_0th column $[c_1, \ldots, c_{k(m)}]$ of the matrix M and suppose that $c_i = c_j = 1$. We may assume without loss of generality that $p \neq j$. Take the codeword a_t for which $a_{ts} = 0$ if $s = 1, \ldots, j-1, j+1, \ldots, m$ and $a_{tj} = 1$. The q_0th verification bit for this codeword must be 1, hence, $a_{tq} = 1$. On the other hand, there certainly exists a codeword a_u for which $a_{up} = a_{uq} = 0$, namely, the null sequence. This shows that columns p and q of the matrix $[a_{ij}]$ cannot be identical or complementary in this case. Suppose, finally, that $p, q > m$. Let $p_0 = p - m$ and $q_0 = q - m$. Consider the p_0th column $[v_1, \ldots, v_{k(m)}]$ and the q_0th column $[w_1, \ldots, w_{k(m)}]$

of the matrix M. Since they are neither identical nor complementary, let i be the position where they coincide and j where they differ. Take a codeword a_t for which $a_{ts} = 0$ if $s = 1, \ldots, i-1, i+1, \ldots, m$ and $a_{ti} = 1$ and a codeword a_u for which $a_{us} = 0$ if $s = 1, \ldots, j-1, j+1, \ldots, m$ and $a_{uj} = 1$. Clearly, the p_0 th and q_0 th verification bits coincide in the first of those words and differ in the second, that is $a_{tp} = a_{tq}$ and $a_{up} \neq a_{uq}$. This shows, however, that columns p and q of the matrix $[a_{ij}]$ cannot be identical or complementary in this case as well; hence, our claim is proved.

We now show that a matrix B satisfying properties 1-4 can always be constructed. If $2m \geq k(m)$ this is obvious: as the first $k(m) - m$ rows take the square matrix $[b_{ij}]$ for which $b_{ij} = 0$ iff $i = j$; complete the remaining rows arbitrarily using distinct binary vectors of weight ≥ 2. Hence, suppose $2m < k(m)$ and let $r = k(m) - m$. By definition r is the least integer s for which $m \leq 2^s - s - 1$. It follows that

$$r > m > 2^{r-1} - (r-1) - 1$$

which implies

$$2^{r-2} < r$$

and, hence, $r \leq 3$, thus, $m \leq 2$, contradiction. Thus, our proof is complete. ∎

It should be noted that for $n = 2^m$, $m \geq 3$, the expression $\min\{k\colon k < 2^{k-m}\}$ from Theorem 2 coincides with the expression $\min\{k\colon n(k+1) \leq 2^k\}$, for even n, in Theorem 1. Thus, for such n, non-interactive non-repetitive search requires no more queries than interactive search with repetitions permitted — if at most one error is allowed.

It remains to consider the case $m = 2$, left out in Theorem 2. Thus, $n = 2^m = 4$. Exhaustive verification shows that no set of 5 non-interactive non-repetitive tests is sufficient to perform search in the presence of one error. On the other hand, 6 tests are clearly enough: $\{1,2\}, \{1,3\}, \{1\}, \{2\}, \{3\}, \{4\}$, is an example.

Thus, the case $m = 2$ (that is, $n = 4$) is an exception from the rule: if non-repetitive non-interactive search is required, one more test has to be used than in the case of non-repetitive interactive search or non-interactive search with repetitions allowed.

Conclusions.

We described optimal non-repetitive searching strategies assuming one error: for any search space in the case of interactive testing and for search spaces of size 2^m in the case of non-interactive testing. We are unable to provide such optimal strategies for an arbitrary fixed bound on the number of errors or even to decide if forbidding repetitions influences the efficiency of questioning in the general case.

In the non-interactive setting the problem has an equivalent formulation with purely combinatorial flavour: find a minimal size family of pairwise non complementary subsets of $\{1,\dots,n\}$ such that each pair of integers $i \leq i < j \leq n$ is separated by at least $2k+1$ sets of the family. This is clearly equivalent to an optimal non-repetitive non-interactive searching strategy assuming k errors.

It is obvious that many Hamming codes, even perfect ones, involve repetitions. Reversing our argument used in the proof of Theorem 2 it is easy to show that if the submatrix B of the generating matrix $M = [\frac{B}{I}]$ contains identical columns then the resulting code yields repetitive tests. Thus, for example, the matrix

$$M = \begin{bmatrix} 0011 \\ 1100 \\ 1110 \\ 1101 \\ 1111 \\ --- \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$$

generates a (perfect) (7,4) code which is not suitable for our purpose.

Let us finally observe that Theorem 1 and Theorem 2 remain true if only identical tests are forbidden and complementary ones allowed. This would be an appropriate assumption if distinct processors were responsible for providing answers to distinct queries, even complementary. The maximal number of faulty units possible to handle in this setting is $2^{n-2} - 1$. The problem of finding an optimal non-interactive questioning strategy assuming k faulty processors has a particularly elegant combinatorial formulation in this case: find a minimal size family of subsets of $\{1,\dots,n\}$ such that every pair $1 \leq i < j \leq n$ is separated by at least $2k+1$ of them. This problem is open even for $k = 1$.

References

1. J. Czyzowicz, D. Mundici, A. Pelc, *Ulam's searching game with lies*, Journal of Combinatorial Theory, Series A **52** (1989), 62–76.
2. W. Guzicki, *Ulam's searching game with two lies*, Journal of Combinatorial Theory, Series A **54** (1990), 1–19.
3. F.J. MacWilliams, N.J.A. Sloane, "The Theory of Error — Correcting Codes", Elsevier Science Publishers B.V., New York, 1977.
4. A. Pelc, *Solution of Ulam's problem on searching with a lie*, Journal of Combinatorial Theory, Series A **44** (1987), 129–140.

5. A. Pelc, *Searching with known error probability*, Theoretical Computer Science **63** (1989), 185–202.
6. A. Pelc, *Detecting errors in searching games*, Journal of Combinatorial Theory, Series A **51** (1989), 43–54.
7. A. Rényi, *On a problem of information theory*, in "Selected Papers of Alfred Rényi", (Pál Turán, ed.) Vol. 2, Akademiai Kiado, Budapest, 1976, pp. 631–638.
8. R.L. Rivest, A.R. Meyer, D.J. Kleitman, K. Winklmann, and J. Spencer, *Coping with errors in binary search procedures*, Journal of Computer and System Sciences **20** (1980), 396–404.
9. J. Spencer, *Guess a number—with lying*, Mathematics Magazine 57 **2** (1984), 105–108.
10. S.M. Ulam, "Adventures of a Mathematician", Scribner's, New York, 1976.