Optimization Letters
https://doi.org/10.1007/s11590-024-02120-1

ORIGINAL PAPER

Cooperation in combinatorial search

Dániel Gerbner[1] · Balázs Keszegh[1,2] · Kartal Nagy[2] · Balázs Patkós[1] · Gábor Wiener[3]

Received: 3 May 2023 / Accepted: 15 April 2024

Abstract
In the game theoretical approach of the basic problem in Combinatorial Search an adversary thinks of a defective element d of an n-element pool X, and the questioner needs to find x by asking questions of type is $d \in Q$? for certain subsets Q of X. We study cooperative versions of this problem, where there are multiple questioners, but not all of them learn the answer to the queries. We consider various models that differ in how it is decided who gets to ask the next query, who obtains the answer to the query, and who needs to know the defective element by the end of the process.

Keywords Cooperative search · Combinatorial search · Search game

1 Introduction

In Combinatorial Search Theory, the basic problem is the following. We are given an n-element underlying set X that contains an unknown defective element d (we assume $n \geq 2$, otherwise all search problems are straightforward). We have to identify the defective element by asking queries of type "is $d \in Q$?" for certain subsets $Q \subseteq X$ (we refer to such a query simply as asking Q). Our aim is of course to use as few queries as possible in order to identify d. What is the minimum number of questions needed and how can we find the queries corresponding to this minimum?

This basic question is easy to answer if we can ask any subset of X: we need $\lceil \log n \rceil$ queries (here and throughout the paper $\log n$ denotes $\log_2 n$). There are several variants that lead to interesting questions: there might be a restriction on the query sets (only sets of size at most k are allowed [7], the defective element is one edge of a graph and edge sets of paths can be queried [2, 8]), some answers can

✉ Kartal Nagy
kartal97@student.elte.hu

1 Alfréd Rényi Institute of Mathematics, Budapest, Hungary

2 ELTE Eötvös Loránd University, Budapest, Hungary

3 Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Budapest, Hungary

Published online: 03 May 2024

be erroneous deliberately [9] or randomly [13], the queries have to be asked at the same time or in a certain number of batches (non-adaptive and k-round algorithms [4, 15]), there might be more defectives, and so on; see the monograph [5] for more details.

Here we introduce a new family of variants, which involve cooperation. Rather than one participant asking the queries and trying to identify the defective element, we are dealing with more participants that have to work together. Each participant receives only some of the information, which is a rather natural assumption. The participants together receive all the information. However, we do not allow outside communication, the only allowed action of the players is to ask a query (when it is their turn) or announce the identification of the defective element. This is another reasonable assumption, since establishing an additional communication channel may be costly or insecure.

We consider different models within this framework. It is unclear what model could be interesting for applications, thus we study basic ones of theoretical interest. The models differ mainly in the information the participants obtain after a query. We use the game theoretic approach of Search Theory (see [1]), where the search process is considered to be a game between Player A (the adversary) and Player B (the questioner) and extend it by adding more players.

Our starting point is a nice puzzle by Sándor Róka [3] (motivated by his research in [10, 11]). Player A picks the defective element and answers the queries, Player B asks the queries (all at the same time, non-adaptively) and Player C is given the YES answers (along with the sets to which these answers correspond, of course). In other words, Player B chooses a family $\mathcal{F}$ of subsets of X, and Player A gives the subfamily $\mathcal{F}'$ to Player C, where $\mathcal{F}' = \{F \in \mathcal{F} : \ d \in F\}$. Upon receiving the answers, Player C has to identify d, which can be done if and only if there exists a unique element of X contained in each member of $\mathcal{F}'$.

It is not hard to see that Player C can identify d if and only if $\mathcal{F}$ is *completely separating*, i.e., $\mathcal{F}$ satisfies the following property: for any $x, y \in X$, there exists a set $F \in \mathcal{F}$ such that $x \in F$, $y \notin F$. The smallest possible size of a completely separating family was determined by Spencer [12]. As we will use its ideas later, we briefly describe the simple proof.

The *dual* of a family $\mathcal{F}$ of subsets of X is defined as a family $\mathcal{H}$ of cardinality $|X|$ on the underlying set $\mathcal{F}$ as follows. For each $x \in X$, we define a set $H_x \in \mathcal{H}$ such that for each $F \in \mathcal{F}$ we have $F \in H_x$ if and only if $x \in F$. It is easy to see that $\mathcal{F}$ is completely separating if and only if its dual is a *Sperner system*, i.e., there are no H_x and H_y in $\mathcal{H}$, such that $H_x \subset H_y$. The classical theorem of Sperner [14] states that the largest possible cardinality of a Sperner system on an m-element underlying set is $\binom{m}{\lfloor m/2 \rfloor}$. This implies that the smallest possible size of a completely separating family on an n-element underlying set is the smallest m such that $\binom{m}{\lfloor m/2 \rfloor} \geq n$.

Let us now turn to new models. In each of the models, Player A picks the defective element and answers the queries. Some models will require adaptive, others require non-adaptive algorithms. Models also differ by numerous other properties, like

- Whether the players can agree in a strategy beforehand,
- Whether just one player can ask queries or more, and in the latter case, what is the order in which the players are allowed to ask their queries,
- What information the players obtain after a query,
- Which players should be able to identify the defective element.

We assign numbers to the models and say that an algorithm solves Model i if the required players can always identify the defective element. If such an algorithm exists, the minimum number of queries needed for the solution (in the worst case, of course) is denoted by $f_i(n)$. Since all models we consider are harder to solve than the basic model of search, the trivial lower bound $f_i(n) \geq \lceil \log n \rceil$ holds for all i.

We start with some models where Player B asks the queries, and multiple players obtain different pieces of information from the answers. These models are non-adaptive, and we assume that the players other than Player B do not have any information beforehand about the algorithm: they are only given a family of sets with the information whether they contain the defective element.

For each model we state our results and prove these later in the next section.

Model 1

We have four players: A, B, C, D. Player B asks the queries non-adaptively, Player C obtains the YES answers and Player D obtains the NO answers (together with the corresponding queries). At least one of C and D have to be able to identify the defective element.

In order to find the optimum strategy for Model 1, we use the notion of k-Sperner systems. A family is said to be k-Sperner if it does not contain $k+1$ distinct sets $S_1, \dots, S_{k+1}$ such that $S_1 \subsetneq S_2 \subsetneq \cdots \subsetneq S_{k+1}$. A theorem of Erdős [6] states that the largest possible cardinality of a k-Sperner family on an m-element underlying set is $\sum_{i=1}^{k} \binom{m}{\lfloor \frac{m-k}{2} \rfloor + i}$, from which the second statement of the following Proposition follows.

Proposition 1.1 *A family $\mathcal{F}$ solves Model 1 if and only if the dual $\mathcal{H}$ of $\mathcal{F}$ is a 2-Sperner family. Therefore,* $f_1(n) = \min \left\{ m : \binom{m}{\lfloor m/2 \rfloor} + \binom{m}{\lfloor m/2+1 \rfloor} \geq n \right\}$.

Our next model is a generalization of Model 1.

Model 2

We have $r+2$ players, $A, B, C_1, \dots, C_r$ with $r > 1$. Player B asks a partition of X to r sets $X_1, \dots, X_r$ as a query. The defective element is contained in X_i for some i; then Player C_i is given a YES answer (together with what the query was). We also say that X_i belongs to Player C_i. The goal is that at least one of Players C_j should be able to identify the defective element. Note that Model 2 is a generalization of Model 1, indeed, since a query F of B in Model 1 corresponds to the query $(F, X \setminus F)$ of B in Model 2.

It is more convenient to state our result on $f_2(n)$ (or, more precisely, on $f_2(n, r)$) in the following form. Let $g_2(k, r)$ denote the largest number of elements such that

k queries are enough to solve Model 2 on $g_2(k, r)$ elements. Then clearly (as in the previous models) $f_2(n, r)$ is the smallest k such that $n \leq g_2(k, r)$.

$$g_2(k,r) = \begin{cases} r\binom{k}{\lfloor k/2 \rfloor} & \text{if } k \geq 3 \text{ and } r \neq 2, \\ \binom{k}{\lfloor k/2 \rfloor} + \binom{k}{\lfloor k/2+1 \rfloor} & \text{if } r = 2. \\ 2r-1 & \text{if } k = 2. \end{cases}$$

Theorem 1.2

The next model differs from Model 2 only in that Player B can ask r disjoint sets that do not need to form a partition. This also means that $r = 1$ makes sense now and in this case we obtain the original puzzle we have dealt with earlier.

Model 3

We are given $r+2$ players, $A, B, C_1, \ldots, C_r$. Player B asks r disjoint subsets $X_1, \ldots, X_r$ of X. If the defective element is contained in X_i for some i, then Player C_i obtains a YES answer (together with what the query was). The goal is that at least one of Players C_i should be able to identify the defective element.

Theorem 1.3 $g_3(k, r) = r\binom{k}{\lfloor k/2 \rfloor}$.

Let us turn our attention to some completely different models. From now on we will have multiple questioners and the goal is to minimize the total number of queries they ask. This makes sense only if they can affect each other, thus here we deal with adaptive search. In the sequel we assume that Players B and C can discuss their (questioning) strategies before asking any queries (but any communication between them is forbidden during the algorithm, of course). We consider only YES-NO queries, where only the YES answer is shared with some player. We also assume that this player does not know whether a query with a NO answer was asked (otherwise they would know that the answer is NO just from hearing no answer). Since such a player may need to ask the next query, the order in which the players ask the queries should be hidden from some players. There is an easy way to achieve this: Player A decides which of the other players will ask the next query.

Model 4

We have three players A, B, C. Player A decides who asks next. Whenever Player B asks a query, if the answer is YES, Player C obtains what the query was (and that the answer was YES). Similarly, whenever C asks a query, if the answer is YES, B obtains what the query was (and that the answer was YES). Note that the player asking the query does not get any information concerning this particular query. The goal is that at least one of B and C should be able to identify the defective element.

Theorem 1.4 $f_4(n) \leq \lceil \log n \rceil + \lceil \log \log n \rceil + 2$.

Model 5

We have three players A, B, C. Player A decides who asks next. Whenever Player B asks a query, B obtains the answer and if the answer is YES, C also obtains what the query was (and that the answer was YES). Similarly, whenever C asks a query, C obtains the answer and if the answer is YES, Player B also obtains what the query was (and that the answer was YES). The goal is that at least one of B and C should be able to identify the defective element. Note that Model 5 differs from Model 4 in that the player asking the query always obtains the answer.

Proposition 1.5 $f_5(n) \leq \lceil \log n \rceil + 1$.

This upper bound is only one larger than the trivial lower bound. For small values of n, we determined the exact value of $f_5(n)$, and it turns out that both $\lceil \log n \rceil$ and $\lceil \log n \rceil + 1$ might occur as such, but we were not able to recognize a pattern. It seems to be an interesting question to determine the set of those n's for which $f_5(n) = \lceil \log n \rceil$.

Model 6

This model is the same as Model 5 with only one difference: the goal is that both B and C should be able to identify the defective element.

Theorem 1.6 $f_6(n) \leq \lceil \log n \rceil + 2\lceil \sqrt{\log n} \rceil + 2$.

Finally, we introduce a third type of models. In these we handle a problem we have mentioned earlier in a different way: if a player obtains information about a query only if the answer is YES, then in the case of a NO answer, they should not know what the query was. Now we handle this problem by assuming that the elements are indistinguishable before the start of the algorithm. When the players discuss their strategy beforehand, in the previous models, they could consider elements $x_1, \dots, x_n$ and choose, say, $\{x_1, \dots, x_{n/2}\}$ as the first query of Player B and $\{x_{n/2+1}, \dots, x_n\}$ as the first query of Player C. Now all they know is that there are n elements in the underlying set. They can still decide that the first query is a set of order $n/2$, but they cannot pick, say, complement sets in this model. Basically, we remove an unfounded assumption about the knowledge the players have concerning the underlying set. The result of this change is that even if Player B knows that the answer was NO to a query asked by Player C, this information is much less useful than before.

Model 7

The elements are indistinguishable. Player B and Player C alternate asking queries. From here we follow Model 4: Whenever Player B asks a query, if the answer is YES, Player C obtains what the query was. Similarly, whenever Player C asks a query, if the answer is YES, Player B obtains what the query was. Note that the player asking the query does not obtain any information. The goal is that at least one of Player B and Player C should be able to identify the defective element. We remark that if the answer is NO to a query asked by (say) B, then C finds out that the answer was NO, since he has to ask the next query without obtaining a YES answer. However, Player C does not find out what the query of Player B was (this is why in this model we need indistinguishable elements, otherwise the two players could emulate the classic non-adaptive search strategy using at most $\lceil \log n \rceil + 1$ steps).

Model 8

Model 8 is the same as Model 7, except that the player asking a query also obtains the answer to the query. Clearly $f_8(n) \leq f_7(n)$.

Recall that the golden ratio is $\phi = \frac{1+\sqrt{5}}{2}$.

Theorem 1.7 $\log_\phi n - O(1) \leq f_8(n) \leq f_7(n) \leq \lceil \log_\phi(n) \rceil$.

2 Proofs

Let us start with the proof of Proposition 1.1. Recall that it states that a family $\mathcal{F}$ solves Model 1 if and only if its dual $\mathcal{H}$ is 2-Sperner.

Proof of Proposition 1.1 Let us assume that $\mathcal{F}$ solves Model 1. If Player C can find the defective element d, then for every element $x \neq d$, there exists a set $F \in \mathcal{F}$ with $d \in F$, $x \notin F$. If Player D can find the defective element, then for every element $x \neq d$, there is a set F' with $x \in F'$, $d \notin F'$. Since any element y can be defective, we have for every y that either for every $x \neq y$, there is a set F with $y \in F$, $x \notin F$, or for every $x \neq y$, there is a set F' with $y \notin F'$, $x \in F'$.

It is equivalent that in the dual family $\mathcal{H}$ for every set H_y, we have that either every other set H_x has the property that there is an F with $F \in H_y$, $F \notin H_x$, or every set H_x has the property that there is an F' with $F' \notin H_y$, $F' \in H_x$. The first property means that H_y is not contained in any H_x, and the second property means that H_y does not contain any H_x. In other words, H_y is either maximal or minimal with respect to containment. Three distinct sets with $H \subset H' \subset H''$ in $\mathcal{H}$ would mean that H' violates this property, thus $\mathcal{H}$ is 2-Sperner.

Observe that in each step of the above proof the implications go both ways, giving us the other direction. Finally, the result on $f_1(n)$ follows from the theorem of Erdős on k-Sperner families, mentioned in the introduction. □

We prove Theorems 1.2 and 1.3 together, as parts of their proofs are common. Observe that by definition we have $g_3(k) \geq g_2(k)$.

Proof of Theorems 1.2 and 1.3 Let us start with proving the general upper bound $r\binom{k}{\lfloor k/2 \rfloor}$. Observe that if $\mathcal{F}$ is a solution to Model 2 or 3, then for every $x \in X$ we have at least one player C_i such that if x is the defective element, then C_i can identify it. Let Y_i be the set of elements that can be identified by C_i.

We claim that $|Y_i| \leq \binom{k}{\lfloor k/2 \rfloor}$. Let $x, y \in Y_i$, then there is a query where $x \in X_i, y \notin X_i$, otherwise if x is the defective, then both x and y appear in all the sets X_i where Player C_i obtains any information, thus Player C_i does not know whether x or y is the defective element. This means that the sets X_i restricted to Y_i form a completely separating family $\mathcal{F}_i$. Therefore, the dual family of $\mathcal{F}_i$ is a Sperner family of cardinality $|Y_i|$ on the underlying set $\mathcal{F}_i$ with $|\mathcal{F}_i| = k$, completing the proof of the claim.

Summing this we get that the number of elements is indeed at most $\sum |Y_i| \le r\binom{k}{\lfloor k/2 \rfloor}$.

Let us continue with the proof of the upper bound in Theorem 1.2 in the case $k = 2$. Assume on the contrary that $g_2(2,r) = 2r$, in this case by the above claim we must have $|Y_i| = 2$ for each i and these sets are pairwise disjoint. Let $X_1, \dots, X_r$ be the first query and $X'_1, \dots, X'_r$ be the second query. Let $Y_1 = \{x, y\}$, then clearly $x \in X_1, y \notin X_1$ and $y \in X'_1, x \notin X'_1$ (or the other way around) because of the completely separating property. As we deal with Model 2, another set in the first query, say X_2, contains y. Let $Y_2 = \{u, v\}$ with $u \in X_2$. Then $u \neq y$ as $Y_1 \cap Y_2 = \emptyset$. If u is the defective, then Player C_2 cannot rule out the possibility that y is the defective, since Player C_2 only obtains the information that X_2 contains the defective (recall that in Model 2 and 3 the Players C_i are not aware of the queries of Player B, they only get some sets that contain the defective element), a contradiction. This completes the proof of the upper bound $g_2(2,r) \le 2r - 1$.

Let us continue with the lower bound $r\binom{k}{\lfloor k/2 \rfloor}$ in the case of Theorem 1.3. Consider a set X of size $r\binom{k}{\lfloor k/2 \rfloor}$. Player B first partitions X to r sets $Y_1, \dots, Y_r$, each of size $\binom{k}{\lfloor k/2 \rfloor}$. On each Y_i, Player B picks a completely separating family of order k and let $X^i_1, \dots, X^i_k$ be its members. Then as Query Q_j, Player B asks the sets X^i_j. If the defective element d belongs to Y_i, then C_i can identify d, as any element not in Y_i is distinguished by any X^i_j that contains d (and there is at least one such set) and any $y \neq d$ in Y_i is distinguished from d as $\mathcal{F}_i$ is completely separating.

The above family of queries solves Model 3 but it cannot be used for Model 2, since the queries do not form a partition of the underlying set. Let us prove now the lower bound $r\binom{k}{\lfloor k/2 \rfloor}$ in the case of Theorem 1.2, $r > 2$ and $k > 2$. Again, on each Y_i Player B picks a completely separating family $\mathcal{F}_i = \left\{X^i_1, \dots, X^i_k\right\}$, but now we specify that the dual of $\mathcal{F}_i$ is the family of $\lceil k/2 \rceil$-element sets of the k element base set. Queries $Q_1, Q_2, \dots, Q_k$ obtained as above are still not partitions of X, so we have to modify them.

There are some elements in Y_i that are not in X^i_j. Let us first add these elements to X^ℓ_j for some $\ell \neq i$. Repeating this for every i, we obtain a new query Q'_j that is a partition of X to $U^1_j, \dots, U^r_j$. Let us repeat this for every j. Then, as all elements of $U^i_j \setminus X^i_j$ are not in Y_i, the sets $U^i_1, \dots, U^i_k$ still form a completely separating family restricted to Y_i. If Player C_i would know that the defective element is in Y_i, then he could identify it. However, it is possible that for some $x \in Y_i$ and $y \notin Y_i$, each set from $\{U^i_1, \dots, U^i_k\}$ that contains x also contains y. If that is the case and x is the defective, then Player C_i cannot rule out the possibility that y is the defective.

Observe that each element of Y_i appears in $\lceil k/2 \rceil$ sets X^i_j by the choice of $\mathcal{F}_i$. Therefore, such an element $x \in Y_i$ appears in $\lceil k/2 \rceil$ sets U^i_j. If $y \notin Y_i$, then $y \in Y_\ell$ for some ℓ, thus such a y appears in $\lceil k/2 \rceil$ sets U^ℓ_j, thus y appears in at most $\lfloor k/2 \rfloor$ sets U^i_j. Thus, if k is odd, then we cannot have that each set U^i_j that contains x also contains y (as $\lfloor k/2 \rfloor < \lceil k/2 \rceil$), which completes the proof if k is odd.

If k is even, then we still want that every element of Y_ℓ appears in less than $k/2$ sets U^i_j for every $\ell \neq i$. To do this, we extend the sets X^i_j to U^i_j more carefully. For each j, we take the elements of $Y_i \setminus X^i_j$, and add each of them to one of two different sets U^ℓ_j and $U^{\ell'}_j$ in the following way. For each element $y \in Y_i$, we first consider the smallest j such that $y \notin X^i_j$, and then we add y to U^ℓ_j. Then we take the second smallest index j' such that $y \notin X^i_{j'}$, and then we add y to $U^{\ell'}_{j'}$. Adding y to these two sets alternately this way, any $y \notin Y_i$ appears in less than $k/2$ sets U^i_j. Therefore, we cannot have that each set U^i_j that contains x also contains y, as every $x \in Y_i$ appears in $k/2$ sets U^i_j. We can execute this plan if $r \geq 3$ (to pick i, ℓ and ℓ') and $k > 2$ (to pick ℓ and ℓ' among the $\lceil k/2 \rceil$ sets not containing y).

The case $r = 2$ is dealt with in Proposition 1.1.

It is left to show that $g_2(2, r) \geq 2r - 1$. Let $X = \{x_1, \dots, x_{2r-1}\}$.

We define the queries $Q_1 = \left(X^1_1, X^1_2, \dots, X^1_r\right)$, $Q_2 = \left(X^2_1, X^2_2, \dots, X^2_r\right)$ by $X^1_i = \{x_{2i-1}\}, X^2_i = \{x_{2i}\}$ for $i \leq r - 1$, and $X^1_r = X \setminus \cup_{i=1}^{r-1} X^1_i$, $X^2_r = X \setminus \cup_{i=1}^{r-1} X^2_i$. If the defective element is x_j with $j < 2r - 1$, then it belongs as a singleton to a Player, thus that Player can identify it. If the defective element is x_{2r-1}, then Player C_r can identify it, since x_{2r-1} is the only element contained in both X^1_r and X^2_r. □

In multiple proofs next, Players B and C can decide their strategy before the algorithm starts. They will apply modifications of the following strategy that we will call the *Basic Strategy*. Let us identify elements of X with 0-1 sequences of length $\lceil \log n \rceil$, and let S_i denote the set of sequences whose ith bit is 1. In the Basic Strategy, the ith query of Player B is S_i, and the ith query of Player C is $\overline{S_{\lceil \log n \rceil - i+1}}$. In other words, Player B asks whether the ith bit is 1, while Player C asks whether the ith bit from the other direction is 0.

The main advantage of a predetermined strategy is that whenever a YES answer is obtained to a query of B, then C obtains this answer and also knows what the earlier queries of B were. Since C knows when the answers were YES, he knows the answer to all the queries of B (and vice versa). The reason that C asks the complements is that in this way when they ask about the same bit, one of them obtains the answer YES, thus one of them obtains the information.

Recall that Players B and C are not allowed to communicate during the algorithm. However, they can send messages to each other via asking certain queries. One simple way to send such a message is to ask X as a query (which clearly would not serve any other purpose). Since the answer to this is YES, the other player will obtain this message in the models studied. One can ask the same query multiple times, thus any message could be sent. These messages will be added at some points during the Basic Strategy.

We continue with the proof of Proposition 1.5. Recall that it deals with the model where Player A decides who asks the next query, the Questioner obtains the answer, while the other Player obtains the answer (and the query) if the answer is YES. The statement is that at least one player can identify the defective element after $\log n + 1$ queries.

Proof of Proposition 1.5 Players B and C apply the Basic Strategy. Then at query $\log n + 1$ they arrive to the same bit i. Let us assume without loss of generality that the defective element is in S_i, thus the answer to the query by Player B is YES. Both players know this answer. No matter which one of them arrived to the last query, that player knows the answer to every earlier query, thus can identify the defective element. Indeed, he knows the answer to his own queries by the properties of Model 5, and knows the answer to the queries asked by the other player, since he knows that the other player asked the queries till the ith bit, and knows which of those queries were answered YES. $\square$

The inequality in this Proposition is not sharp. We determined the exact number of questions for values of n up to 16. We found no obvious pattern, thus we do not have any conjecture about the exact value of $f_5(n)$.

$$f_5(n) = \begin{cases} 1 & \text{if } n = 2 \\ 2 & \text{if } n = 3 \\ 3 & \text{if } 4 \le n \le 6 \\ 4 & \text{if } 7 \le n \le 11 \\ 5 & \text{if } 12 \le n \le 16. \end{cases}$$

In the case of 4 elements, two queries are not enough. Both players have to decide on a query, they ask as their first query Q if they have not gained any information earlier, i.e., either Q is the very first query, or the other player asked some queries earlier but they were answered NO. If $|Q| \neq 2$, then the answer could be that there are at least 3 possible defective elements, thus clearly one more query is not enough to solve the problem. If both players have a 2-element set as the first query, Player A let them both ask a query and answers NO, then none of them knows the answer unless the two queries are complements of each other, in which case Player A can answer NO to the first query and YES to the second one with both Players B and C not knowing the defective element.

Next, we show that three queries are enough if there are (at most) six elements. The first query of Player B is $\{1, 2, 3\}$, and the first query of Player C is $\{1, 4, 5\}$ if they do not get any information from the queries of the other player. If they still do not get any other information after this query and the answer is NO, Player B continues with $\{4, 6\}$, and Player C continues with $\{2, 6\}$. If the answer is YES, Player B asks $\{1, 2\}$, and Player C asks $\{1, 4\}$.

Without loss of generality, the first query is by Player B, thus it is $\{1, 2, 3\}$. If the answer is YES, then Player B and C can finish with 2 more queries, as $f_5(3) = 2$.

If the answer to $\{1, 2, 3\}$ is NO and Player B is next, he asks $\{4, 6\}$, in the case of NO, Player B knows the defective; in the case of YES, both players know that 4 or 6 is the defective element and can ask $\{4\}$.

If the answer to $\{1, 2, 3\}$ is NO and Player C is next, he has to ask $\{1, 4, 5\}$. If the answer to that is YES, then Player B knows that the defective is 4 or 5, so he can finish with one more query if he asks the next query. If Player C asks the third query, he has to ask $\{1, 4\}$, so the YES answer reveals to Player B that 4 is the defective element, while the NO answer reveals to Player C that 5 is the defective element.

If the answer to $\{1, 4, 5\}$ is also NO, we are in a symmetric situation, so we can assume that Player B is next. He has to ask $\{4, 6\}$. A NO answer reveals to Player B that 5 is the defective element, while a YES answer reveals to Player C that 4 is the defective element.

Next, we show that 3 queries are not enough if $n = 7$. No matter what the first query is, a possible answer gives at least 4 options for the defective element, thus at least $f_5(4) = 3$ further queries are needed.

Next, we show that four queries are enough if there are (at most) eleven elements.

We provide a strategy for Players B and C to determine the defective element out of 11 possibilities with four queries. Let $X_1 = \{1, 2, 3, 4, 5, 6\}$ and $X_2 = \{7, 9, 11\}$ be the first two queries for Player B if they get to ask them without any knowledge about queries from Player C. Similarly, let $Y_1 = \{1, 2, 3, 7, 8, 9\}$ and $Y_2 = \{4, 6, 10\}$ be the first two queries for Player C if they get to ask them without any knowledge about queries from Player C. Observe that Player A cannot answer all of X_1, X_2, Y_1, Y_2 with NO, as these sets cover all elements. If Player B (or Player C) is allowed to ask a query after receiving NO answers to X_1, X_2 (or Y_1, Y_2), then after the next query, $\{8\}$ (or $\{5\}$), they will know the defective element. Therefore, we can assume that Player A answers a query with YES among X_1, X_2, Y_1, Y_2. If X_1, Y_1 are answered with NO and X_2 with YES, then Player C knows the answer to all the queries (as he received the answer to X_2, so X_1 must have been asked before) and can determine that 11 is the defective element. If X_1, X_2 received no answers and Y_1 received a YES, then Player B received all the answers and can determine that 8 is the defective element. With symmetry, the above cases cover all possibilities when the YES answer came with two or three NO answers.

If the first answer is YES, then Players B and C follow the strategy above for six elements with three queries. So we can assume that X_1 is the first query and answered with NO, while Y_1 is the second query and answered with YES. Now Player C must play according to the winning strategy on six elements, as he does not know about the query X_1. Note that the first two queries for Player C in the above strategy, in case of a first NO answer, form a disjoint pair of a 3-set and a 2-set. So Player C picks $Y_2' = \{1, 2, 7\}$ and $Y_3' = \{3, 9\}$. On the other hand, Player B can change his strategy. So if he receives no further information, he asks $X_3' = \{7, 8\}$. If Player B ever asks X_3' and receives a NO answer, he knows that the defective element is 9. Suppose Player B gets a YES to X_3'. If this was the third query, then whoever asks the fourth query will know the defective element after that. If Y_2' was answered with NO, and then X_3' was answered with YES, then Player C knows that the defective element is 8. So we can assume that X_3' is never queried. If Y_2' is answered with YES, then Player B knows that the defective element is 7. If Y_2', Y_3' are both answered with

NO, then Player C knows that the defective element is 8. Lastly, if Y_2' is answered with NO and Y_3' answered with YES, then Player B knows that the defective element is 9. This finishes the proof of $f_5(11) \leq 4$.

Finally, we show that in the case of 12 elements, Players B and C need at least 5 queries. Then $f_5(n) = 5$ will follow for $12 \leq n \leq 16$ since for these values of n, we have $\lceil \log n \rceil + 1 = 5$. Suppose, towards a contradiction, that Players B and C have a strategy to determine the defective element with at most 4 queries. Let X_1, X_2 (Y_1, Y_2) be the first two queries Player B (Player C) asks if he does not know anything else than the answers to his own queries, i.e., all queries (if any) of the other Player were answered with NO, and in the case of the second query, the first answer to their own query was NO. Also, let X_2' (Y_2') be the query that Player B (Player C) asks if all he knows is that the answer to X_1 (Y_1) was YES, and so all previous queries of Player C (Player B) were answered with NO.

If X_1 or Y_1 has a size different from 6, then if Player A lets this query be asked first, there is a way to answer such that the number of remaining possible defective elements is at least 7 (in the case this comes with a NO answer, Player A can additionally reveal this information to all other players). This is a contradiction as $f_5(7) = 4$. So from now on, we will assume $|X_1| = |Y_1| = 6$.

If $|X_1 \cap Y_1| \neq 3$, then one of $\overline{X}_1 \cap Y_1$ or $\overline{X}_1 \cap \overline{Y}_1$ has size at least 4, then Player A can let X_1 be the first query and answer it NO, then let Y_1 be the second query and answer it YES or NO so that there will remain at least 4 possible defective elements, which is a contradiction as $f_5(4) = 3$. So from now on, we will assume $|X_1 \cap Y_1| = |\overline{X}_1 \cap Y_1| = |X_1 \cap \overline{Y}_1| = |\overline{X}_1 \cap \overline{Y}_1| = 3$. Similarly, if one of $|\overline{X}_1 \cap X_2|$ or $|\overline{X}_1 \cap \overline{X}_2|$ ($|\overline{Y}_1 \cap Y_2|$ or $|\overline{Y}_1 \cap \overline{Y}_2|$) is at least 4, then Player A can let X_1, X_2 (Y_1, Y_2) be the first two queries and answer X_1 NO and X_2 so that the number of remaining possible defective elements is at least 4, which is again a contradiction. So from now on, we assume $|\overline{X}_1 \cap X_2| = |\overline{X}_1 \cap \overline{X}_2| = |\overline{Y}_1 \cap Y_2| = |\overline{Y}_1 \cap \overline{Y}_2| = 3$.

If $X_2 \supset \overline{X}_1 \cap \overline{Y}_1$ ($Y_2 \supset \overline{X}_1 \cap \overline{Y}_1$), then Player A can let X_1, Y_1, X_2 (X_1, Y_1, Y_2) be the first three queries and answer them NO, NO, YES. The number of remaining possible defective elements is at least 3, which is a contradiction as $f_5(3) = 2$. So from now on, we can and will assume that $|X_2 \cap \overline{X}_1 \cap \overline{Y}_1| \leq 2$ and $|Y_2 \cap \overline{X}_1 \cap \overline{Y}_1| \leq 2$.

By symmetry, we assume $X_1 = \{7, 8, \ldots, 12\}$ and $Y_1 = \{4, 5, \ldots, 9\}$. Suppose $|X_2 \cap \{1, 2, 3\}| \leq 1$, say $X_2 \cap \{1, 2, 3\}$ is empty or the singleton $\{1\}$. Then Player A can let the first four queries be X_1, Y_1, X_2, Y_2 and answer the first three of them NO, while the answer to Y_2 is YES if Y_2 contains $\{2, 3\}$ and NO otherwise. If the answer was YES, then both Player B and Player C know that the defective element is one of 2 and 3, but a further query is needed to determine which one. If the answer was NO, then both Player B and Player C know the answers to their own queries, an they both have see three possible defectives. Note that knowing all the answers may be enough to figure out that the defective element (depending on whether $X_2 \cap \{1, 2, 3\}$ is empty or not), but none of the players knows all the answers.

In both cases, the strategy needs at least 5 queries. A similar argument shows that the strategy cannot work if $|Y_2 \cap \{1, 2, 3\}| \leq 1$, so we can assume that

$|X_2 \cap \{1,2,3\}| = |Y_2 \cap \{1,2,3\}| = 2$ and thus $|X_2 \cap Y_1 \cap \overline{X}_1| = |Y_2 \cap X_1 \cap \overline{Y}_1| = 1$. In particular, we assume $X_2 = \{1,2,4\}$.

We are left with two subcases. Suppose first that $|Y_1 \cap Y_2'| \geq 5$. Then Player A lets Player C ask the first two queries and answers YES for both of them. As a consequence, Player C asked Y_1 and Y_2', so after two queries both Players B and C know that the defective element is in $Y_1 \cap Y_2'$, which is a contradiction as $f_5(5) = 3$, so three more queries have to be asked.

Suppose next $|Y_1 \cap Y_2'| \leq 4$. Then Player A lets Player B ask the first two queries and answers both of them NO (and so the queries are X_1, X_2). Then, Player A lets Player C ask the next two queries and answers first YES, and so the queries are Y_1 and Y_2'. If $Y_2' \supset 5,6$, then Player A answers YES to Y_2'; otherwise, NO. Observe first that if $Y_2' \not\supset 5,6$, then a NO answer is possible since $X_1 \cup X_2 \cup Y_2'$ does not cover Y_1. If the answer was YES, then both 5 and 6 are possible defective elements. If the answer is NO, then for Player C, any element in $Y_1 \setminus Y_2'$ is a possible defective element, and as $|Y_1 \cap Y_2'| \leq 4$, there are at least two of them. While Player B only knows the answers to X_1, X_2 and Y_1 and so 5 and 6 are still possible defective elements for him. This contradiction finishes the proof of $f_5(12) \geq 5$.

Let us continue with the proof of Theorem 1.6. Recall that Model 6 differs from Model 5 in that both players should be able to identify the defective element. The bound we have to prove is $f_6(n) \leq \lceil \log n \rceil + 2\lceil \sqrt{\log n} \rceil + 2$.

Proof of Theorem 1.6 Let P be the set of numbers of the form $i\lceil \sqrt{\log n} \rceil$ for $i \leq \lceil \sqrt{\log n} \rceil$. Players B and C follow the Basic Strategy with some additional queries. For simplicity, we describe the strategy for Player B; it is analogous for Player C. Whenever Player B asks S_p with $p \in P$, the next query of Player B is a *special query*. Afterward Player B continues with the Basic Strategy, i.e., her next query is S_{p+1}. The special query is X, thus the answer is always YES, thus Player C obtains this information. Therefore, Player C also knows which special query this was, thus Player C knows the answer to each S_i with $i < p$.

Whenever a query corresponding to the ith bit is asked by both players, one of them obtains a YES answer, thus the other player knows every bit, as in the proof of Proposition 1.5. Assume now that the special query at $p \in P$ is asked by both players. Then Player B knows the answer to each S_i with $i \leq p$ since he asked those queries and the answer to each $\overline{S_i}$ with $i \geq p$ since those were asked by Player C, and this special query shows this fact. Player C analogously knows each bit, thus the algorithm is finished.

As there are $\lceil \sqrt{\log n} \rceil$ elements in P, there are at most $\lceil \sqrt{\log n} \rceil + 1$ special queries asked. For each $i \leq \log n$, one of the queries S_i and $\overline{S_i}$ is asked, except for the the values i between two consecutive elements of P. Indeed, if without loss of generality, Player B asks the special query at p first, it is possible that he asks all the queries between p and the next element $p + \lceil \sqrt{\log n} \rceil$ of P before Player C asks the special query at p. However, if Player B would ask also the special query at $p + \lceil \sqrt{\log n} \rceil$, then this special query would be asked by both players before the special query at p, thus the algorithm would finish before Player C would ask the special query at p, leading to a contradiction. Therefore, there are at most $\lceil \sqrt{\log n} \rceil + 1$ entries i with both S_i and $\overline{S_i}$ asked, hence there are $\lceil \log n \rceil + \lceil \sqrt{\log n} \rceil + 1$ non-special queries asked. □

We remark that we could get rid of the +2 term by making the strategy less symmetric: Player B would ask the special query at p after asking S_p and Player C would ask it before asking $\overline{S_p}$.

Let us continue with the proof of Theorem 1.4. Recall that the player asking the query does not get any information, but it is enough if one of Players B and C finds the defective element. The theorem states that $f_4(n) \le \lceil \log n \rceil + \lceil \log \log n \rceil + 2$.

Proof of Theorem 1.4 Players B and C again follow the Basic Strategy with the following alterations. Throughout the process, they maintain a left endpoint ℓ, a right endpoint r, and a target position $t = \frac{\ell+r}{2}$. At the beginning, $\ell = 1$ and $r = \lceil \log n \rceil$. B plays along the Basic Strategy unless he is to ask a query S_i with $i > t$, while C plays along the Basic Strategy unless he is to ask a query $\overline{S_i}$ with $i < t$. At these moments B and C ask a *special query*, for which the answer is always YES and which is of the form $\cap_{i \in I} S_i \cap \cap_{i \in I'} \overline{S_i}$. The special query is the smallest set S the questioner knows the defective element belongs to. Formally, without loss of generality, assume that it is C who asks the special query $Q = Q_1 \cap Q_2$, where Q_1 is the set that C knows because of the previous queries of B, and Q_2 is the set that C knows about his own queries. More precisely, if we denote by YES_B the set of queries asked by B that were answered YES, then

$$Q_1 = \bigcap_{S' \in YES_B} S' \cap \bigcap_{\exists i' > i : S_{i'} \in YES_B} \overline{S_i}.$$

Observe that C does know $\bigcap_{S' \in YES_B} S'$ as these are queries of B that were answered YES, so he received them. He also knows $\bigcap_{\exists i' > i : S_{i'} \in YES_B} \overline{S_i}$ as the players agreed to play along the Basic Strategy, so if C did not receive S_i but received an $S_{i'}$ with $i < i'$, then C knows that the defective element is in $\overline{S_i}$. An important consequence of asking the special query Q that has a part Q_1 is that B after receiving Q will know the answers to all of his previous queries. Indeed, as C asks $\bigcap_{S' \in YES_B} S'$, B knows which of his queries have been answered YES, and thus all his other queries must have been answered NO. If before C is asking a special query, there had been special queries from B, then C knows the answers to those of his queries that were asked before B's last special query, so C can include these S_js and $\overline{S_j}$s into Q_2.

There are two ways how a special query can be asked. The simpler case is if B asks this instead of asking $S_{\lfloor t \rfloor + 1}$ (or similarly C asks this instead of asking $S_{\lceil t \rceil - 1}$). In this case, ℓ is changed to $\lfloor t \rfloor + 1$ and t is changed according to the rule $t = \frac{\ell+r}{2}$. (If C was asking the special query, then r becomes $\lceil t \rceil - 1$ and t is updated accordingly.) Another possibility is that a Player asked a special query and that moved the target t "behind" the other player. For example if B is first to reach the bit $\frac{1}{2} \log n$, but by that time C has already asked $\overline{S_{\frac{5}{8} \log n}}$, then t becomes $\frac{3}{4} \log n > \frac{5}{8} \log n$. This situation can only occur when say, C asks his first query after a special query of B (maybe B has asked queries since, but not C), so as mentioned above, at this moment C knows the answers to all the previous queries by himself. Therefore, the Q_2 part of his special query will reveal his current position i to B, so the two Players can set r to be i and then update t.

As the value of $r - \ell$ is at most half of its previous value after each change to t, we obtain that the number of special queries is at most $\lceil \log(\lceil \log n \rceil) \rceil$. To finish the proof, we need to analyze the situation when B has asked S_i, C has asked $\overline{S_{i+1}}$, so until this point $\lceil \log n \rceil$ normal and at most $\lceil \log(\lceil \log n \rceil) \rceil$ special queries have been asked.

Case I We have $t > i + 1$, i.e., that target is behind C.

Then by the above observation, C knows the answers to all his previous queries. If C is to ask the next query (a special query as $t > i + 1$), then B will learn the answers to all the bits, and thus, he will know defective element. If B is to ask S_{i+1}, then there are two possibilities. If the answer is YES, then C will receive this, and will know the defective element, as because of the Basic strategy, C will know all the YES and thus all NO queries of B. Finally, if the answer to S_{i+1} would be NO, then previously the answer to C's query $\overline{S_{i+1}}$ must have been YES and thus received by B. So B instead of asking S_{i+1} can ask a special query ensuring that C will know the defective element.

Case II We have $t = i + \frac{1}{2}$ or $t = i + 1$.

If C is to ask the next query, then the query is a special one that lets B know the answers to all his queries and moves the target behind B. So any later query from B would be a special query letting C know the defective element. If, following C's special query, C proceeds to ask the next query, it will be $\overline{S_i}$. Then the situation is analogous to the second subcase of Case I: if the answer is YES, B receives this information and discovers the answers to all queries by C. If the answer is NO, C already knows this, as B had asked S_i, so he can ask a special query instead of $\overline{S_i}$.

If B is to ask the next query and $t = i + \frac{1}{2}$, then the roles of B and C are symmetric, so the above paragraph can be applied. Finally, if B is to ask the next query and $t = i + 1$, then after the next query, both players would proceed with a special query, and we finish as before.

In all cases, we finished with at most 2 extra queries. This completes the proof. □

Let us continue with the proof of Theorem 1.7. Recall that now Players B and C can agree in a strategy beforehand, but the elements are indistinguishable at that point. They alternate asking queries, and at least one of them should be able to identify the defective element.

Proof of Theorem 1.7 Recall first that for the golden ratio we have that $1/\phi + 1/\phi^2 = 1$.

Let us start with the upper bound. At each point, one of Players B and C is the *main questioner*, the other is the *auxiliary questioner*. Let us assume that Player B asks the first query, then he starts as the main questioner. They change roles whenever there is a YES answer.

If the answer to the query of the main Questioner is NO, the auxiliary questioner asks the whole underlying set X. This tells the main Questioner that the answer to his Query was NO.

The main questioner always maintains the set of possibly defective elements X_i and asks a subset Y_i of X_i of order $\lfloor |X_i|/\phi \rfloor$ as a query. If the answer is YES, the other player becomes the main questioner, and $X_{i+1} := Y_i$ is the set of possibly defective elements. If the answer is NO, then the auxiliary questioner queries X, and the main Questioner sets the set of possibly defective elements to $X_i \setminus Y_i$. This way, (omitting the floor signs) the size of the set of possibly defective elements either decreases to $|X_i|/\phi$ with one query or decreases to $|X_i|(1 - 1/\phi) = |X_i|/\phi^2$ with two queries. Clearly this implies the upper bound $f_7(n) \leq \log_\phi n$.

Let us continue with the lower bound. We describe the strategy of the adversary recursively. Note that usually an adversary strategy means that we pick the answers so that there is still a possible solution. However, as the elements are indistinguishable, we can also control the intersection of the queries by the two players in the case the second player does not have any information on the query of the first player. For example, if the first player asks a set of size $n/2$ and the answer is NO, and the second player asks a set of size $n/2$, the adversary can decide that they asked the same set.

If the first query has size at least $\lfloor n/\phi \rfloor$, the answer is YES, otherwise NO. If the first answer is YES, he applies the same strategy starting at the next query. If the first answer is NO, then note that the complement of the first query has size at least $\lfloor n/\phi^2 \rfloor$. If the second query has size at least $\lfloor n/\phi^2 \rfloor$, then the adversary decides that the intersection of the second query with the complement of the first query has size at least $\lfloor n/\phi^2 \rfloor$ and the answer is YES. Moreover, he tells both players all the answers to the queries. This way after two queries there are at least $\lfloor n/\phi^2 \rfloor$ possibly defective elements, and the adversary applies the same strategy starting at the next query. On the other hand, if the second query has size less than $\lfloor n/\phi^2 \rfloor$, the adversary answers NO and decides that the union of the two queries has size at most $\lfloor n/\phi \rfloor$ and tells this information to both players. Again, after two queries there are at least $\lfloor n/\phi^2 \rfloor$ possibly defective elements, and the adversary applies the same strategy starting at the next query. This completes the proof. □

3 Concluding remarks

There are other models that one can obtain by combining the ideas used in the paper. Some of these do not make much sense, e.g. if player A decides who asks the next query, and the player asking the query does not get any information, we cannot require that both players identify the defective, since A can make sure that only one player asks all the queries, and that player can obviously never identify the defective.

Other models are just not of much interest, because their solution is trivial. E.g., one might ask what happens in Model 1 if we require that both players can identify the defective. This cannot be easier than the simple puzzle we started with, where Player C has to identify the defective. But the solution to that is a completely separating family, and it is not hard to see that if the queries form a completely separating family, then Player D can also identify the defective element, completely solving this case. Another example is if Players B and C ask queries alternately, the other player obtains the YES answers and the elements are distinguishable. Then the players agree on a strategy and each player knows the answer to the queries of the other player, thus they can always decrease the set of possibly defective elements by a factor of 2, which leads to an algorithm requiring $\log n$ queries.

We never forbade agreeing in a strategy before the start of an adaptive algorithm. The reason is that forbidding this leads to a sort of philosophical question. Can the players agree in a strategy in constant time? They can communicate during the algorithm (by sending messages via asking certain queries) and send any number k to each other, e.g., by asking the query X exactly k times in a row. Without discussing the strategy, but assuming optimal play by the players, can we assume that the other player understands that the message is k and can he decode this message? If yes, then forbidding the discussion beforehand only gives a constant additive term.

In this paper we restricted our study to worst case bounds. Yet one could also consider the expected length of algorithms in case the defective element is chosen randomly (and then we can also decide if Player A should choose randomly which Player can ask a query next in the appropriate models, or this could be the choice of A).

Let us finally remark that while cooperation was essential in our algorithms, we could have studied competitive versions of Combinatorial Search instead, using the same models. For example, in Models 4,5,6,7,8 it could be the goal of Player B that Player C should not be able to identify the defective element (and vice versa). What are the chances that Player B wins when the defective is a (uniformly) random element and how many queries are needed in the optimal algorithm? In Models 1,2,3 the goal of Player B could be e.g. that one player cannot identify the defective element but multiple players can.

Funding Open access funding provided by Eötvös Loránd University. Research of Gerbner is supported by the National Research, Development and Innovation Office - NKFIH under the Grants SNN 129364, FK 132060, and KKP-133819. Research of Keszegh is supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, by the National Research, Development and Innovation Office – NKFIH under the Grant K 132696 and FK 132060, by the ÚNKP-21-5 and ÚNKP-22-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund and by the ERC Advanced Grant "ERMiD". Research of Patkós is supported by the National Research, Development and Innovation Office - NKFIH under the Grants SNN 129364 and FK 132060. This research has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 and the TKP2021 (project no. BME-NVA-02) funding scheme.

Data availability We do not analyse or generate any datasets, because our work is theoretical. One can obtain the relevant materials from the references below.

References

1. Aigner, M.: Combinatorial Search. Wiley, Hoboken (1988)
2. Bonamy, M., Botler, F., Dross, F., Naia, T., Skokan J.: Separating the edges of a graph by a linear number of paths. (2023). arXiv preprint arXiv:2301.08707
3. Birkás, G.y., Hraskó, A.: Personal communication (2022)
4. D'yachkov, A.G., Vorobyev, I.V., Polyanskii, N.A., Shchukin, V.Y.: On a hypergraph approach to multistage group testing problems. In 2016 IEEE International Symposium on Information Theory (ISIT), pp. 1183–1191. IEEE, (2016)
5. Du, D.-Z., Hwang, F.K.: Combinatorial Group Testing and Its Applications. World Scientific, Singapore (1999)
6. Erdős, P.: On a lemma of littlewood and offord. Bull. Am. Math. Soc. **51**, 898–902 (1945)
7. Katona, G.: On separating systems of a finite set. J. Combin. Theory **1**, 174–194 (1966)
8. Letzter, S.: Separating paths systems of almost linear size. arXiv preprint arXiv:2211.07732 (2022)
9. Pelc, A.: Searching games with errors-fifty years of coping with liars. Theoret. Comput. Sci. **270**(1–2), 71–109 (2002)
10. Róka, S.: Independent intersection systems (in Hungarian). Acta Acad. Paedag. Nyíregyháziensis **12**, 17–20 (1990)
11. Róka, S.: Independent intersection systems II. Acta Acad. Paedag. Agriensis, Sect. Math. **24**, 67–73 (1997). (**in Hungarian**)
12. Spencer, J.: Minimal completely separating systems. J. Combin. Theory **8**, 446–447 (1970)
13. Spencer, J., Winkler, P.: Three thresholds for a liar. Comb. Probab. Comput. **1**(1), 81–93 (1992)
14. Sperner, E.: Ein Satz über Untermengen einer endlichen Menge. Math. Z. **27**, 544–548 (1928)
15. Vorobyev, I.: Optimal multistage group testing algorithm for 3 defectives. In 2020 IEEE International Symposium on Information Theory (ISIT), pp. 90–95. IEEE, (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.