

Алгоритм адаптивного группового тестирования

К.Ю. Горбунов

Институт проблем передачи информации им. А.А. Харкевича, Российская академия наук, Москва,
Россия

Поступила в редколлегию 10.04.2025 г. Принята 30.04.2025 г.

Аннотация—Описывается алгоритм группового тестирования в условиях, когда в данном множестве мощности n имеется неизвестное количество $d > 0$ дефектных элементов и требуется их идентифицировать последовательностью тестов. Каждый следующий тест может учитывать результаты предыдущих. Верхняя оценка на число тестов в нашем алгоритме улучшает известную оценку. Время алгоритма минимально возможное — порядка числа тестов.

КЛЮЧЕВЫЕ СЛОВА: групповое тестирование, адаптивное тестирование, алгоритм тестирования, поиск дефектов, идентификация инфицированных.

DOI: 10.53921/18195822_2025_25_1_56

1. ВВЕДЕНИЕ

В задачах группового тестирования требуется идентифицировать все дефектные элементы (в медицинской терминологии — инфицированные индивиды) в данном множестве S мощности n , используя минимальное количество тестов, т.е. запросов о наличии хотя бы одного дефектного элемента в тестовом подмножестве. Эти задачи различаются по двум критериям: известно ли заранее число d инфицированных индивидов (или оценки на него) и разрешается ли проводить тесты последовательно (адаптивное тестирование) или только параллельно. Параллельное тестирование рассмотрено в работах [1], [2], [3], где приводятся дальнейшие ссылки. Последовательному тестированию посвящена статья [4], где приводятся многочисленные ссылки по истории задачи. Результат этой статьи мы собираемся улучшить.

Итак, рассматриваем случай адаптивного тестирования, когда число $d > 0$ заранее не известно, а тесты проводятся последовательно, один за другим. В этих условиях в [4] получена оценка $d \log \frac{n}{d} + (5 - \log 5)d + O(\log^2 d)$ (все логарифмы здесь и далее берутся по основанию 2). В этой задаче ключевой вопрос состоит в понижении коэффициента при втором слагаемом. Для неё мы опишем алгоритм с лучшей верхней оценкой $d \log \frac{n}{d} + (7 - \log 21)d + O(\log^2 d)$. Общая схема нашего алгоритма такая же, как в [4], доказательство оценки несколько проще. В описании алгоритма мы сохраняем, насколько возможно, терминологию и обозначения из [4].

2. ОПИСАНИЕ АЛГОРИТМА

Алгоритм использует две вспомогательных процедуры: DIG и 4-Split, а также числовую последовательность: $a_0 = 1$, $a_1 = 2$, $a_2 = 4$, $a_3 = 6$ и $a_i = 11 \cdot 2^{i-4}$ для $i > 3$. На вход каждой процедуры подаётся множество X , о котором известно, что оно содержит хотя бы один дефектный элемент.

Процедура DIG. Эта процедура решает задачу идентификации одного дефектного элемента в X . Пока $|X| > 1$ выбираем в X произвольное подмножество Y мощности $\lceil \frac{|X|}{2} \rceil$ (верхняя

целая часть), тестируем его и, если результат положительный (т.е. Y содержит дефектный элемент), полагаем $X = Y$, а иначе $X = X \setminus Y$. Если в X остаётся один элемент (напомним, $d > 0$), идентифицируем его как дефектный. Индукцией по $|X|$ легко доказать, что число тестов в DIG равно $\log |X|$, если $|X|$ — степень двойки, и не превышает $1 + \log |X|$ иначе. Заметим, что когда мы получаем в DIG отрицательный результат, мы не считаем элементы соответствующего множества идентифицированными.

Процедура 4-Split. Она получает на вход, кроме X , параметр — натуральное число v , для которого $|X| \leq a_v$. Если $|X| \leq 4$, применяем DIG к X . Если $|X| = 5$ или $|X| = 6$, выбираем в X произвольное подмножество X' мощности 2 и тестируем его. Если результат положительный, полагаем $X = X'$, иначе идентифицируем все элементы из X' как недефектные и полагаем $X = X \setminus X'$. Затем применяем к X процедуру DIG. Если $|X| \geq 7$, произвольно разбиваем X на 4 множества Y, Z, U, V (некоторые из них могут быть пустыми), где $|Y| = \min\{2^{v-2}, |X|\}$, $|Z| = \min\{3 \cdot 2^{v-4}, |X| - |Y|\}$, $|U| = \min\{2^{v-3}, |X| - |Y| - |Z|\}$, $|V| = |X| - |Y| - |Z| - |U|$ (очевидно, $|V| \leq 2^{v-3}$). Если Z пусто, применим DIG к Y . Иначе тестируем множества Y, Z, U, V в этом порядке до первого положительного результата, идентифицируя элементы множеств с отрицательным результатом как недефектные (если первые три теста отрицательны, четвёртый не проводится). Обозначим X' первое “дефектное” множество.

Если $X' = Y$, $X' = U$ или $X' = V$, применим DIG к X' . Если $X' = Z$, выбираем в Z произвольное подмножество Z' мощности $\min\{|Z|, 2^{v-4}\}$ и тестируем его. Если результат положительный, полагаем $X = Z'$, иначе идентифицируем все элементы из Z' как недефектные и, если $Z \setminus Z'$ непусто, полагаем $X = Z \setminus Z'$. Затем применяем к X процедуру DIG. Отметим: смысл процедуры 4-Split в том, чтобы либо уменьшить область поиска больше, чем вдвое, либо получить недефектные идентифицированные элементы.

Основной алгоритм. На каждой итерации образуется текущее множество S , которому приписан некоторый натуральный ранг k . На первой итерации полагаем S равным исходному S и его ранг равным минимальному k такому, что $n \leq a_k$. Итерации выполняются, пока S непустое.

Опишем содержание итерации. Дано множество S и его ранг k . Выберем произвольное подмножество S' в S мощности a_k , если возможно, или иначе положим $S' = S$. Тестируем S' . Если результат отрицательный, идентифицируем все элементы из S' как недефектные, удалим их из S , увеличим k на 1 и перейдём к следующей итерации. Если результат положительный, выполним следующее. Если $k > 0$, применим к S' процедуру 4-Split с параметром $v = k$, удалим из S идентифицированные дефектный и недефектные элементы, уменьшим k на 1 и перейдём к следующей итерации. Если $k = 0$, то S' содержит ровно один элемент. Идентифицируем его как дефектный и удалим из S , оставим $k = 0$ и перейдём к следующей итерации.

Будем называть итерацию *положительной* или *отрицательной* в соответствии с результатом тестирования S' .

3. ОЦЕНКА ЧИСЛА ТЕСТОВ В АЛГОРИТМЕ

Теорема. Число тестов в описанном алгоритме не превосходит

$$d \log \frac{n}{d} + (7 - \log 21)d + \frac{\log^2 d}{2} + 2 \log d + 3.$$

Доказательство.

Лемма 1. Число тестов в процедуре 4-Split с параметром v не превышает v . Если в результате выполнения процедуры 4-Split все элементы из X идентифицированы, то число тестов в ней не превышает $2 + \log |X|$. Число тестов в последней итерации алгоритма

не превышает $3 + \log |X|$. Ранги всех итераций не превосходят ранга k первой итерации, который не больше $\log n + 5 - \log 11$.

Доказательство. Обозначим $N(m)$ число тестов в 4-Split на множестве X мощности m . Очевидно $N(1) = 0$ ($v \geq 0$), $N(2) = 1$ ($v \geq 1$), $N(3) = N(4) = 2$ ($v \geq 2$), $N(5) \leq 3$ ($v \geq 3$), $N(6) \leq 3$ ($v \geq 3$) и для $m \leq 6$ первое утверждение выполнено. Если $m \geq 7$, рассмотрим случаи.

Случай 1. Подмножество Y дефектное. Поскольку $|Y| \leq 2^{v-2}$, выполняется $N(m) \leq 1 + (v - 2) = v - 1$.

Случай 2. Подмножество Y недефектное, Z' дефектное. Поскольку $|Z'| \leq 2^{v-4}$, выполняется $N(m) \leq 3 + (v - 4) = v - 1$.

Случай 3. Подмножества Y и Z' недефектные, $Z \setminus Z'$ дефектное. Поскольку $|Z \setminus Z'| \leq |Z| - 2^{v-4} \leq 3 \cdot 2^{v-4} - 2^{v-4} = 2^{v-3}$, выполняется $N(m) \leq 3 + (v - 3) = v$.

Случай 4. Подмножества Y и Z недефектные, U дефектное. Поскольку $|U| \leq 2^{v-3}$, выполняется $N(m) \leq 3 + (v - 3) = v$.

Случай 5. Подмножества Y , Z и U недефектные, V дефектное. Поскольку $|V| \leq 2^{v-3}$ и V не тестируется, выполняется $N(m) \leq 3 + (v - 3) = v$.

Второе утверждение леммы при $|X| = 1$ очевидно. В случае $|X| \geq 2$ заметим, что последнее непустое множество из Y , Z' , $Z \setminus Z'$, U , V одноэлементно, поэтому число тестов в процедуре не больше 3. Третье утверждение следует из второго. Первая часть последнего утверждения леммы следует из того, что итерация ранга k действует на множестве S' , содержащем все неидентифицированные элементы из S , и потому положительна или является последней. Неравенство $k \leq \log n + 5 - \log 11$ при $n \leq 11$ проверяется непосредственно, а при других n следует из того, что $n \geq a_{k-1}$. $\square$

Обозначим $I_1, I_2, \dots, I_q$ итерации алгоритма. Как и в [4], разбиваем множество итераций на три класса: C_1 , C_2 и C_3 . Обозначим r_i — ранг итерации I_i , t_i — число тестов на ней, n_i — число идентифицированных на ней (дефектных и недефектных) элементов. Класс C_1 содержит положительные итерации ранга 0 и итерацию I_q . Класс C_2 содержит итерацию I_1 и для каждого натурального числа r , $r_q < r < r_1$, первую в последовательности $I_1, I_2, \dots, I_q$ итерацию ранга r . Класс C_3 содержит все остальные итерации.

Лемма 2. Число итераций из C_3 чётно, и их можно разбить на такие пары, что ранги итераций из каждой пары отличаются ровно на 1 (будем обозначать эти ранги как v и $v-1$), итерация ранга v положительна, итерация ранга $v-1$ отрицательна, и соответствующее ей подмножество S' имеет мощность ровно a_{v-1} .

Лемма 2 совпадает с леммой 3.5 из [4], поэтому её доказательство не приводится. $\square$

Фиксируем указанное в лемме 2 разбиение на пары, называемые *зигзаг-парами*. Будем обозначать зигзаг-пару как (I_i, I_j) , где ранг итерации I_i равен v , а ранг итерации I_j равен $v-1$. Соответствующее число v называем *рангом пары*.

Лемма 3. Для каждой зигзаг-пары $P = (I_i, I_j)$ выполняется неравенство

$$t_i + t_j \leq \log(n_i + n_j) + 7 - \log 21.$$

Доказательство. Пусть $v \geq 1$ — ранг пары P . Для $v = 1$ имеем $t_i + t_j = 3$, $n_i + n_j = 2$, откуда следует утверждение леммы. Для $v = 2$ имеем $t_i + t_j = 4$, $n_i + n_j = 3$ и утверждение леммы. Для $v = 3$ имеем $t_i + t_j = 4$, $n_i + n_j = 5$ или $t_i + t_j = 5$, $n_i + n_j = 7$ и утверждение леммы. При $v \geq 4$ процедура 4-Split применяется в итерации I_i и разбивает соответствующее множество S'_i на четыре части Y , Z , U , V . Рассмотрим случаи.

Случай 1. Подмножество Y дефектное. Поскольку $|Y| \leq 2^{v-2}$, выполняется $t_i + t_j \leq 3 + (v - 2) = v + 1$. Поскольку $|S'_j| = a_{v-1} = 11 \cdot 2^{v-5}$, имеем $n_i + n_j \geq 11 \cdot 2^{v-5}$, откуда следует $v + 1 \leq \log(n_i + n_j) + 6 - \log 11$ и утверждение леммы.

Случай 2. Подмножество Y недефектное, Z' дефектное. Поскольку $|Z'| \leq 2^{v-4}$, выполняется $t_i + t_j \leq 5 + (v - 4) = v + 1$. Тогда $n_i + n_j \geq 2^{v-2} + 11 \cdot 2^{v-5} = 19 \cdot 2^{v-5}$, откуда следует $v + 1 \leq \log(n_i + n_j) + 6 - \log 19$ и утверждение леммы.

Случай 3. Подмножества Y и Z' недефектные, $Z \setminus Z'$ дефектное. Поскольку $|Z \setminus Z'| \leq |Z| - 2^{v-4} \leq 3 \cdot 2^{v-4} - 2^{v-4} = 2^{v-3}$, то $t_i + t_j \leq 5 + (v - 3) = v + 2$. Тогда $n_i + n_j \geq 2^{v-2} + 2^{v-4} + 11 \cdot 2^{v-5} = 21 \cdot 2^{v-5}$, откуда следует $v + 2 \leq \log(n_i + n_j) + 7 - \log 21$ и утверждение леммы.

Случай 4. Подмножества Y и Z недефектные, U дефектное. Поскольку $|U| \leq 2^{v-3}$, то $t_i + t_j \leq 5 + (v - 3) = v + 2$. Тогда $n_i + n_j \geq 2^{v-2} + 3 \cdot 2^{v-4} + 11 \cdot 2^{v-5} = 25 \cdot 2^{v-5}$, откуда следует $v + 2 \leq \log(n_i + n_j) + 7 - \log 25$ и утверждение леммы.

Случай 5. Подмножества Y , Z и U недефектные, V дефектное. Поскольку $|V| \leq 2^{v-3}$ и V не тестируется, то $t_i + t_j \leq 5 + (v - 3) = v + 2$. Тогда $n_i + n_j \geq 2^{v-2} + 3 \cdot 2^{v-4} + 2^{v-3} + 11 \cdot 2^{v-5} = 29 \cdot 2^{v-5}$, откуда следует $v + 2 \leq \log(n_i + n_j) + 7 - \log 29$ и утверждение леммы. $\square$

Обозначим T_i ($i = 1, 2, 3$) число тестов, выполняемых на всех итерациях из C_i .

Лемма 4. Пусть x ($0 \leq x \leq d$) — число элементов, идентифицированных как дефектные на итерациях из $C_1 \cup C_3$. Если $x = 0$, то $|T_1 \cup T_3| = 1$. Иначе $|T_1 \cup T_3| \leq 1 + (7 - \log 21)x + x \log \frac{n}{x}$.

Доказательство. Если $x = 0$, ранги итераций монотонно уменьшаются от r_1 до r_q , и все итерации, кроме отрицательной итерации I_q , лежат в C_2 , откуда следует первое утверждение. Для доказательства второго утверждения рассмотрим случаи.

Случай 1. Итерация I_q отрицательна. Обозначив как (I_i, I_j) зигзаг-пару, по лемме 3 получаем:

$$|T_1 \cup T_3| \leq |C_1| + \sum_{(I_i, I_j)} (\log(n_i + n_j) + 7 - \log 21) = |C_1| + \frac{|C_3|(7 - \log 21)}{2} + \sum_{(I_i, I_j)} \log(n_i + n_j).$$

Заметим, что $x = |C_1| - 1 + \frac{|C_3|}{2}$. Добавим к полученному выражению нулевое слагаемое $(|C_1| - 1) \log 1$, чтобы число логарифмических слагаемых было равно x , а сумма аргументов логарифмов не превышала n . Используя выпуклость логарифмической функции и неравенство $|C_1| \geq 1$, получим:

$$\begin{aligned} |T_1 \cup T_3| &\leq |C_1| + \frac{|C_3|(7 - \log 21)}{2} + x \frac{\sum_{(I_i, I_j)} \log(n_i + n_j) + (|C_1| - 1) \log 1}{x} \leq \\ &\leq |C_1| + \frac{|C_3|(7 - \log 21)}{2} + x \log \frac{n}{x} \leq 1 + (7 - \log 21)x + x \log \frac{n}{x}. \end{aligned}$$

Случай 2. Итерация I_q положительна. По леммам 1 и 3 получим:

$$\begin{aligned} |T_1 \cup T_3| &\leq |C_1| + \sum_{(I_i, I_j)} (\log(n_i + n_j) + 7 - \log 21) \leq \\ &\leq (|C_1| - 1) + 3 + n_q + \frac{|C_3|(7 - \log 21)}{2} + \sum_{(I_i, I_j)} \log(n_i + n_j) = \\ &= 2 + |C_1| + \frac{|C_3|(7 - \log 21)}{2} + \sum_{(I_i, I_j)} \log(n_i + n_j). \end{aligned}$$

Заметим, что $x = |C_1| - 1 + \frac{|C_3|}{2}$. Добавим к полученному выражению нулевое слагаемое $(|C_1| - 1) \log 1$. Как и выше, получим:

$$\begin{aligned} |T_1 \cup T_3| &\leq 2 + |C_1| + \frac{|C_3|(7 - \log 21)}{2} + x \frac{\sum_{(I_i, I_j)} \log(n_i + n_j) + (|C_1| - 1) \log 1 + \log n_q}{x} \leq \\ &\leq 2 + |C_1| + \frac{|C_3|(7 - \log 21)}{2} + x \log \frac{n}{x} \leq 1 + (7 - \log 21)x + x \log \frac{n}{x}. \quad \square \end{aligned}$$

Лемма 5. Пусть x ($0 \leq x \leq d$) — число элементов, идентифицированных как дефектные на итерациях из $C_1 \cup C_3$, k — ранг итерации I_1 . Тогда $|T_2| \leq \frac{((2k+3)-(d-x))(d-x)}{2}$.

Лемма 5 совпадает с леммой 3.10 из [4], поэтому её доказательство не приводится. $\square$

Лемма 6. При $1 \leq x \leq d$ выполняется неравенство $x \log x \geq d \log d - (\log d + \log e)(d - x)$.

Доказательство. Рассмотрим функцию $f(y) = (d - y) \log(d - y)$ при $0 \leq y \leq d - 1$ и продифференцируем её дважды: $f'(y) = -\log(d - y) - \log e$, $f''(y) = \frac{\log e}{d - y}$. Поскольку вторая производная всюду положительна, приближение функции $f(y)$ касательной в точке $y = 0$ не превышает значения $f(y)$. Отсюда $d \log d - x \log x \leq d \log d - (d \log d - (\log d + \log e)(d - x))$, откуда следует утверждение леммы. $\square$

Теорему сначала проверим для $x = 0$. По леммам 1, 4 и 5 получим:

$$\begin{aligned} |T_1 \cup T_2 \cup T_3| &\leq 1 + \frac{((2k+3)-d)d}{2} \leq \frac{(2(\log n + 5 - \log 11) + 3 - d)d}{2} = \\ &= 1 + d \log n + (6.5 - \log 11)d - 0.5d^2. \end{aligned}$$

Требуется показать, что последнее выражение не превышает $d \log n - d \log d + (7 - \log 21)d + 3$, т.е. что $0.5d^2 - d \log d + (0.5 - \log 21 + \log 11)d + 2 \geq 0$. Поскольку $0.5 - \log 21 + \log 11 \geq -0.5$, достаточно показать, что $0.5d^2 - d \log d - 0.5d + 2 \geq 0$. Проверим это, исследовав на минимум функцию $g(d) = 0.5d^2 - d \log d - 0.5d$ при натуральных d . Имеем $g'(d) = d - \log d - \log e - 0.5$, $g''(d) = 1 - \frac{\log e}{d}$, откуда следует, что $g(d)$ возрастает при $d \geq 4$. Таким образом, достаточно вычислить минимум из $g(1)$, $g(2)$, $g(3)$ и $g(4)$, который, как легко видеть, равен $g(4) = -2$.

Пусть $x \geq 1$. По леммам 1, 4 и 5 имеем:

$$\begin{aligned} |T_1 \cup T_2 \cup T_3| &\leq 1 + (7 - \log 21)x + x \log n - x \log x + \frac{(2(\log n + 5 - \log 11) + 3 - (d - x))(d - x)}{2} = \\ &= -0.5(d - x)^2 + (\log 21 - \log 11 - 0.5)(d - x) + (\log n + 7 - \log 21)d + 1 - x \log x \leq \\ &\leq -0.5(d - x)^2 + 0.5(d - x) + (\log n + 7 - \log 21)d + 1 - x \log x. \end{aligned}$$

По лемме 6 последнее выражение не превышает

$$-0.5(d - x)^2 + (\log d + \log e + 0.5)(d - x) + (\log n + 7 - \log 21)d - d \log d + 1.$$

Сумма первых двух слагаемых — квадратичная функция $g(y) = -0.5y^2 + (\log d + \log e + 0.5)y$ от $y = d - x$. Оценим её значение в точке максимума: $g'(y) = -y + \log d + \log e + 0.5 = 0$, $y_{\max} = \log d + \log e + 0.5$,

$$\begin{aligned} g(y_{\max}) &= -0.5(\log d + \log e + 0.5)^2 + (\log d + \log e + 0.5)^2 = 0.5(\log d + \log e + 0.5)^2 = \\ &= 0.5 \log^2 d + (\log e + 0.5) \log d + 0.5 \log^2 e + 0.5 \log e + \frac{1}{8} \leq 0.5 \log^2 d + 2 \log d + 2. \end{aligned}$$

Получим $|T_1 \cup T_2 \cup T_3| \leq d \log \frac{n}{d} + (7 - \log 21)d + \frac{\log^2 d}{2} + g(y_{\max}) + 1$, что и завершает доказательство теоремы. $\square$

4. ЗАКЛЮЧЕНИЕ

Мы уменьшили константу c в оценке $d \log \frac{n}{d} + cd + O(\log^2 d)$ числа тестов в алгоритме адаптивного группового тестирования до $c = 7 - \log 21$. При этом время алгоритма минимально возможное — порядка числа тестов. Представляет интерес вопрос о дальнейшем уменьшении этой константы, возможно, за счёт увеличения времени работы алгоритма. Также, интересно было бы получить нижнюю оценку на число тестов.

Работа выполнена в рамках государственного задания ИППИ РАН, утверждённого Минобрнауки России.

СПИСОК ЛИТЕРАТУРЫ

1. Coja-Oghlan A., Gebhard O., Hahn-Klimroth M. and Loick P. Optimal group testing. *Combinatorics, Probability and Computing*, 2021, pp. 1–28.
2. Bshouty N.H. Improved lower bound for estimating the number of defective items. *Journal of Combinatorial Optimization*, 2025, vol. 49, art. 33.
3. Coja-Oghlan A., Hahn-Klimroth M., Hintze L., Kaaser D., Krieg L., Rolvien M. and Scheftelowitsch O. Noisy group testing via spatial coupling. *Combinatorics, Probability and Computing*, 2025, vol. 34, pp. 210–258.
4. Wu J., Cheng Y., Du D-Z. An improved zig zag approach for competitive group testing. *Discrete Optimization*, 2022, vol. 43, art. 100687.

An Algorithm For Adaptive Group Testing

K.Yu Gorbunov

We describe an algorithm for group testing in conditions where in the set of cardinality n there is an unknown amount of $d > 0$ defective elements and it is necessary to identify them with a sequence of tests. Each next test can take into account the results of the previous ones. The upper estimate for the number of tests in our algorithm improves the known value. The algorithm runtime is the minimum possible: it is of the order of number of tests.

KEYWORDS: group testing, adaptive testing, algorithm of testing, search of defects, identification of infected individuals.