Combinatorial group testing algorithms improved for *d* = 3

ANTONÍN JANČAŘÍK
Faculty of Education
Charles University
M. Rettigove 4 116 39 Praha 1
CZECH REPUBLIC

Abstract: This paper aims to improve one well-known method for $d = 3$. In the original article, two algorithms were presented, one for $d = 3$ and another (Chinese remainder sieve method) that was adjustable for arbitrary d. In its basic form, the Chinese remainder sieve method was always better than the explicit algorithm for $d = 3$. In our proposed form, the modified algorithm for $d = 3$ is more efficient for some small n, and it also pushes the lower bound from which an efficient algorithm exists.

Key-Words: combinatorial group testing, three defective items, pooling, one-round algorithms, various radixes, three defective samples, speed of testing, scalability of testing

Received: September 15, 2022. Revised: October 18, 2023. Accepted: November 11, 2023. Published: December 7, 2023.

1 Introduction

Combinatorial group testing is a method that efficiently tests many individuals for diseases like COVID-19 by pooling and testing their samples ([1], [2], [3]). This approach conserves testing resources and increases the speed and scalability of testing. The idea of group testing, based initially on health care needs, has proven to be applicable in many other fields such as genetics (e.g., [4]), computer science (e.g., [5], [6]), and engineering (e.g., [7], [8]).

The practical implementation of a method is often limited by testing capabilities, such as the number of samples that can be tested simultaneously or the number of tests that can be performed from a single sample. This limitation necessitates the development of methods that can effectively handle small sample sizes. This paper presents an enhancement of a well-known method for d=3 ([9]).

Two algorithms were introduced in the original paper ([9]). In its basic form, the Chinese remainder sieve method consistently outperformed the explicit algorithm for $d = 3$. However, in our proposed approach, the modified algorithm for $d = 3$ exhibits greater efficiency for some small values of n, thereby pushing the lower bound for the existence of an efficient algorithm.

The paper is organized as follows. Our main result – improved algorithm is in section 2, data comparison for small n is in section 3, and the conclusions follow in section 4.

2 Main results

The original algorithm was designed to work only with number notation in the binary system; we will show that it can be applied similarly, or even more easily, in systems with a different basis.

Let the number of items be $n = k^q$, and the item indices be expressed in radix k. Index $X = X_{(q-1)} \dots X_0$, where each digit $X_p \in \{0, 1, \dots, k-1\}$.

Now, X ranges over the item index numbers $\{0, 1, \dots, n-1\}$, p ranges over the radix positions $\{0, 1, \dots, q-1\}$, and v ranges over the digit values $\{0, 1, \dots, k-1\}$.

Matrix M has $k^2\binom{q}{2}$ rows. Row (p, p', v, v') of M is associated with distinct radix positions p and p', where $p < p'$, and with values v and v', each of which is in $\{0, 1, \dots, k-1\}$. $M[((p, p', v, v'), X] = 1$ iff $X_p = v$ a $X_{p'} = v'$.

Let $test_M(p, p', v, v')$ be the result (1 for positive, 0 for negative) of testing items having a 1-entry in row (p, p', v, v') in M. For $p' > p$ define $test_M(p', p, v, v') = test_M(p, p', v, v')$.

The following three functions can be computed in terms of $test_M$.

$test_B(p, v)$ has value 1 (0) if there are (not) any defectives having value v in radix position p, i.e. $test_B(0, v) = 0$ if $\sum_{i=0}^{k-1} test_M(0, 1, v, i) = 0$, and $test_B(0, v) = 1$ otherwise. For $p > 0$, $test_B(p, v) = 0$ if

$$\sum_{i=0}^{k-1} test_M(0, p, i, v) = 0$$

and 1 otherwise.

$test1(p)$ is the number of different values held by defectives in radix position p. Thus $test1(p) = \sum_{i=0}^{k-1} test_B(p, i)$.

$test2(p, p')$ is the number of different ordered pairs of values held by defectives in the designated ordered pair of radix positions. Therefore, $test2(p, p') = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} test_M(p, p', i, j)$.

Now, we determine the number of defective items and the value of their digits.

Let $T = max_{p=0}^{q-1}(test1(p))$.

Lemma 2.1. *If $T = 0$, there are no defective items.*

Proof. Obvious. □

Lemma 2.2. *If $T = 1$ then $test1(p) = 1$ for all p. Denote by X_p the element for which $test_B(p, X_p) = 1$. Then there is just one defective item $X = X_{(q-1)} \dots X_0$.*

Proof. Obvious. □

The following lemma describes a new case and must be added to the original proof.

Lemma 2.3. *If $T = 3$, then there are just three defective items.*

Proof. Because $T = 3$, exists p such that $test1(p) = 3$ and $X_{p_1}, X_{p_2}, X_{p_3}$ such that $test_B(p, X_{p_i}) = 1$. Then for every p' different from p and X_{p_i} exists just one $X_{p'_i}$ such that $test_M(p, p', X_{p_i}, X_{p'_i}) = 1$. Searched defective items have coordinates $X_i = X_{(q-1)_i} \dots X_{0_i}$ ($i \in \{1, 2, 3\}$). □

What remains to be resolved is a case where $T = 2$. Two cases can occur, $max_{p,p'=0}^{q-1}(test2(p, p')) = 2$ and $max_{p,p'=0}^{q-1}(test2(p, p')) = 3$.

Lemma 2.4. *If $T = 2$ and $max_{p,p'=0}^{q-1}(test2(p, p')) = 3$, there are just three defective items.*

Proof. If $max_{p,p'=0}^{q-1}(test2(p, p')) = 3$, it is already straightforward to distinguish all defective items using pair p, p' such that $test2(p, p') = 3$. □

Lemma 2.5. *If $T = 2$ and $max_{p,p'=0}^{q-1}(test2(p, p')) = 2$, there are exactly two defective items.*

Proof. Suppose there are three defective items, and p is such that $test1(p) = 2$. Then one of the defectives (say D) has in p value v, and the other two (say E, F) have a value $u, u \neq v$. Since E and F are distinct, they must differ in value in some other position p'. Therefore, there will be three different order pairs of values held by defectives in positions p and p', so $test2(p, p') = 3$. A contradiction. □

Theorem 2.6. *M is the 3-separable matrix for $n = k^q$ with $k^2\binom{q}{2}$ rows, for any positive integers $k, q, k \geq 2$.*

Proof. It follows directly from the preceding Lemmas. □

3 Comparison of the number of tests required for the *d* = 3 method

First, let us compare the original algorithm with the newly proposed one. The differences are described in Table 1. The NA means $t(n) > n$, so applying the algorithm is ineffective.

The original algorithm was designed only for $k = 2$, but using a different base could be more efficient. The modified algorithm can be applied for smaller n, except for three values of n $n = 126, 127$, and 128; it is always better than the original one. In Table 1, we always list $t(n)$ only for the k that is most efficient in the given interval.

Table 1: Comparing $t(n)$ for $d = 3$.

n	**origin** $d = 3$	**new** $d = 3$
1–47	NA	NA
48–59	NA	48 ($k = 4$)
60–64	60	48 ($k = 4$)
65–81	NA	54 ($k = 3$)
62–83	NA	75 ($k = 5$)
84–125	84	75 ($k = 5$)
126–128	84	84 ($k = 2$)
129–243	112	90 ($k = 3$)
129–243	112	90 ($k = 3$)
244–256	112	96 ($k = 4$)
257–512	144	135 ($k = 3$)
513–729	180	135 ($k = 3$)
730–1024	180	160 ($k = 4$)

It now remains to compare whether, for some small n, the modified algorithm is more efficient than the previously best-presented algorithm – the Chinese remainder sieve method with backtracking search (denoted bktrk) ([9]). A comparison of the algorithms can be found in Table 2.

Table 2: Comparing $t(n)$ for $d = 3$.

n	**bktrk**	**origin** $d = 3$	**new** $d = 3$
1–47	N/A	N/A	N/A ($k = 4$)
48–49	47	N/A	48 ($k = 4$)
50–56	49	N/A	48 ($k = 4$)
57–60	53	N/A	48 ($k = 4$)
61–64	53	60	48 ($k = 4$)
65–71	53	N/A	54 ($k = 3$)
72–77	57	N/A	54 ($k = 3$)
78–79	58	N/A	54 ($k = 3$)
81	59	N/A	54 ($k = 3$)
82	59	N/A	75 ($k = 5$)
83	60	N/A	75 ($k = 5$)
84–100	60	84	75 ($k = 5$)

The original algorithm was worse for all n than the Chinese remainder sieve method algorithm. However, this is no longer true for the modified algorithm. The modified algorithm offers better results for $n = 50 \dots 64$ and $n = 72 \dots 81$. Thus, the results were optimized by adjusting the algorithm in several cases considered the best known.

4 Conclusions

The modification of the algorithm for three defective items presented in this paper brings only minor improvements, yet it provides the best possible solution for a few small values n. The advantage of the modified algorithm is its ease of implementation. It is not without interest that for small values of n, different values of k ($k = 2, 3, 4, 5$) appear to be the most effective. For larger n, in most cases, it is most advantageous to choose $k = 3$. Although still for $n = 3^{15} + 1 \dots 2^{24}$, $k = 4$ is the most effective.

Future research should focus on applying the results presented in this paper to algorithms that run in more than one round (e.g., [10]). The improvements introduced here could help to improve them, especially in future rounds of testing where the number of defective samples in a group is already limited. Finding optimal algorithms for small n using ICT would undoubtedly be interesting.

References:

[1] T. Bardini Idalino, L. Moura, Structure-aware combinatorial group testing: a new method for pandemic screening. In *International Workshop on Combinatorial Algorithms*. Cham: Springer International Publishing, 2022. p. 143–156.

[2] F. Huang, P. Guo, Y. Wang, Optimal group testing strategy for the mass screening of SARS-CoV-2. *Omega*, Vol. 112, 2022, no. 102689.

[3] V. H. da Silva, C. P. Goes, P. A. Trevisoli, R. Lello, L. G. Clemente, T. B. de Almeida, J. Petrini, L. L. Countinho, Simulation of group testing scenarios can boost COVID-19 screening power. *Scientific Reports*, Vol. 12, No. 1, 2022, no: 11854.

[4] M. Farach, S. Kannan, E. Knill, S. Muthukrishnan, Group testing problems with sequences in experimental molecular biology, In: *Proceedings. Compression and Complexity of SEQUENCES 1997*, IEEE, 1997, pp. 357–367

[5] A. De Bonis, G. Di Crescenzo, Combinatorial group testing for corruption localizing hashing, In: *International Computing and Combinatorics Conference*, Springer, Berlin, 2011, pp. 579–591

[6] L. Lazic, S. Ppopovic, N. Mastorakis, A simultaneous application of combinatorial testing and virtualization as a method for software testing. *WSEAS Transactions on Information Science and Applications*, Vol. 6, No. 11, 2009, pp. 1802–1813.

[7] S. W. Chiu, K. K. Chen, J. C. Yang, M. H. Hwang, Deriving the Optimal Production-Shipment Policy with Imperfect Quality and an Amending Delivery Plan using Algebraic Method. *WSEAS Transactions on Systems*, Vol. 11, No. 5, 2012, pp. 163–172.

[8] M. T. Goodrich, D. S. Hirschberg, Improved adaptive group testing algorithms with applications to multiple access channels and dead sensor diagnosis, *Journal of Combinatorial Optimization*, Vol. 15, No. 1, 2008, pp. 95–121

[9] D. Eppstein, M. T. Goodrich, D. S. Hirschberg, Improved combinatorial group testing algorithms for real-world problem sizes, *SIAM Journal on Computing*, Vol. 36, 2007, pp. 1360–1375

[10] T. Mehta, Y. Malinovsky, C. C. Abnet, P. S. Albert, Using group testing in a two-phase epidemiologic design to identify the effects of a large number of antibody reactions on disease risk. *BMC Medical Research Methodology*, Vol. 22, No. 1, 2022, pp. 1–9.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

A. Jančařík – investigation, validation and writing & editing.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.