Cascaded Group Testing

Waqar Mirza
Department of EE
IIT Bombay
Mumbai, India
wmirza608@gmail.com

Nikhil Karamchandani
Department of EE
IIT Bombay
Mumbai, India
nikhilk@ee.iitb.ac.in

Niranjan Balachandran
Department of Mathematics
IIT Bombay
Mumbai, India
niranj@math.iitb.ac.in

Abstract—In this paper, we introduce a variation of the group testing problem where each test is specified by an ordered subset of items, and returns the first defective item in the specified order. We refer to this as *cascaded group testing* and the goal is to identify a small set of K defective items amongst a collection of size N, using as few tests as possible. For the adaptive testing regime, we show that a simple scheme is able to find all defective items in at most K tests, which is optimal. For the non-adaptive setting, we first come up with a necessary and sufficient condition for any collection of tests to be feasible for recovering all the defectives. Using this, we are able to show that any feasible non-adaptive strategy requires at least $\Omega(K^2)$ tests. In terms of achievability, it is easy to show that a collection of $O(K^2 \log(N/K))$ randomly constructed tests is feasible. We show via carefully constructed explicit designs that one can do significantly better. We provide two simple schemes for $K = 1, 2$ which only require one and two tests respectively irrespective of the number of items N. Note that this is in contrast to standard binary group testing, where at least $\Omega(\log N)$ tests are required. The case of $K \geq 3$ is more challenging and here we come up with an iterative design which requires only $\mathrm{poly}(\log \log N)$ tests.

I. INTRODUCTION

The problem of *group testing* was originally formulated in [1], with the motivation being efficient screening of syphilis-infected soldiers during World War II. A mathematical model for the problem entails identifying a subset of K defective items from a population of size N, using as few *pooled tests* as possible. In standard group testing, each pooled test specifies a subset of the items and produces a binary outcome: 'negative' if all the items selected in the test are non-defective, and 'positive' otherwise. While the original motivation for group testing was medical testing, it has since found application across a wide variety of scenarios including wireless communications [2]–[4], DNA sequencing [5], neighbor discovery [6], [7] and network tomography [8], [9]. There is a lot of literature related to the group testing problem and we are only able to discuss a small sample of it below; we refer the interested reader to [5], [10]–[12] for a more comprehensive coverage of the problem as well as known technical results.

Group testing strategies can be divided into two categories: *adaptive* and *non-adaptive*. The former refers to a setting where the configuration of i-th pooled test can be decided based on the outcomes of the previous $i-1$ tests. On the other hand, non-adaptive group testing requires all the tests to be specified beforehand so that they can be potentially conducted in parallel. Since the introduction of the group testing problem, there has been a lot of work on characterizing the optimal number of tests required for identifying K defectives out of N items. A simple counting argument shows that any feasible strategy with T tests should satisfy $2^T \geq \binom{N}{K}$, and thus $T \geq \log_2 \binom{N}{K}$ provides a lower bound. There exist adaptive group testing strategies whose performance is very close to this bound; for example, a generalized binary splitting algorithm in [13] is guaranteed to require at most $\log_2 \binom{N}{K} + K$ tests. Similarly, both lower bounds and efficient strategies have been proposed for the non-adaptive setting as well. For example, [14], [15] showed that any non-adaptive group testing strategy requires at least $\min\{N, \Omega\left(K^2 \log_K N\right)\}$ tests. On the other hand, explicit testing strategies based on coding-theoretic ideas were proposed in [16] and later in [17], which require at most $O\left(K^2 \log_K^2 N\right)$ and $O\left(K^2 \log N\right)$ tests respectively.

Several variants of the binary 'OR' pooled test in standard group testing, as described above, have been studied in the literature. These include *threshold* testing [18], [19] where the outcome is positive only if the number of defective items included in the test is larger than a specified threshold; *quantitative* testing [20] which has a non-binary outcome like the number of defective items included in the test; and *concomitant* testing [21] where a test outcome is positive only if it includes at least one item from each of a pre-defined collection of subsets.

In this work, we study another testing model with non-binary output, which we call *cascade testing*. Each test is specified by an ordered subset of items $S = (i_1, i_2, ..., i_r)$, and returns the position of the first defective item in S according to the specified order. Note that this test provides more information than the standard binary 'OR' test. As motivation for this model, consider a network tomography application where the goal is to identify the congested links using probes. Each probe traverses through a chosen path in the network and either goes through completely, in which case none of the links on the path are congested; or it returns the identity of the first congested link along the path[1]. Another application would be a movie recommendation system which sequentially presents options from different genres to a user, till he/she picks one; and then uses this feedback to learn the subset of genres that the user likes. Such a feedback model has also

---

[1] For example, Simple Mail Transfer Protocol (SMTP) can provide such information while selecting efficient paths for email delivery

recently received significant attention in the Online Learning community under the moniker 'cascading bandits' [22]–[24], with applications in opportunistic spectrum access, network routing, recommendation systems, and dynamic treatment allocation.

The focus of this work is on characterizing the minimum number of tests required to identify K defectives amongst N items, under the cascade testing model. For the adaptive testing regime, we demonstrate a simple scheme that enables us to find all defective items in at most K tests, which is optimal. For the non-adaptive setting, we first come up with a necessary and sufficient condition for any collection of tests (*design*) to be feasible for recovering all the defectives. Using this equivalence, we are able to show that any feasible non-adaptive strategy requires at least $\Omega(K^2)$ tests. In terms of achievability, it is easy to show that a collection of $O(K^2 \log N)$ randomly constructed tests is feasible. We show via carefully constructed explicit designs that one can do significantly better. We provide two simple schemes for $K = 1, 2$ which only require one and two tests respectively irrespective of the number of items N. Note that this is in contrast to standard binary group testing, where at least $\Omega(\log N)$ tests are required. The case of $K \geq 3$ is much more challenging and here we come up with an iterative design which requires order-wise $(\log \log N)^{a_K}$ tests for a fixed K, where a_K is a parameter that depends on K.

The rest of the paper is organized as follows. We provide the problem formulation for cascaded group testing in Section II. The adaptive and non-adaptive settings are considered in Sections III and IV respectively. Bounds on the optimal number of tests needed for non-adaptive group testing are presented in Section V. Finally, some explicit constructions are provided in Section VI and a short discussion is presented in Section VII.

II. PROBLEM FORMULATION

We have a set of N items, denoted by $V = \{v_1, ..., v_N\}$, out of which an unknown subset $\mathcal{K}$ are defective. We assume $|\mathcal{K}| \leq K$ for some known $0 < K \leq N$ and aim to recover $\mathcal{K}$ through 'tests'. In standard binary group testing [1], each test is specified by a subset S of items and returns 1 if S contains a defective item, i.e., $S \cap \mathcal{K} \neq \emptyset$, and returns 0 otherwise.

On the other hand, in this work, we consider an alternate testing model we call *cascaded testing* where a test t is associated with an ordered subset of items $(v_{i_1}, v_{i_2}, ..., v_{i_{|t|}})$, where $|t|$ denotes the number of items involved in the test. The test returns the first defective item in this sequence. In other words, the result of the test is $y = v_{i_r}$ where[2] $r = \min\{j \in [|t|] : v_{i_j} \in \mathcal{K}\}$, assuming there is some defective item in the test. If there are no defective items involved, the test returns[3] $y = 0$. Note that a cascaded test provides at least as much information as a standard binary group test.

The main focus of this work is to characterize the minimum number of tests required in the adaptive and non-adaptive settings under cascaded testing. This involves deriving lower bounds as well as designing achievability schemes.

[2]For $n \in \mathbb{N}$, $[n] := \{1, 2, .., n\}$.
[3]We impose the constraint $0 \notin V$ for consistency.

We introduce some notation for convenience. We say $x \in t$ (resp. $x \notin t$) to mean x is an item involved in (resp. not in) the test t. For $x, y \in t$, we say $x <_t y$ if x appears before y in t, and $x \leq_t y$ if x appears before y or $x = y$. For any set U, let $t \cap U$ $(= U \cap t)$ denote the ordered set intersection of t and U, with the same order of items as in t. Similarly, $t \backslash U := t \cap U^c$. We will sometimes denote the result of the test as $y = \min(t \cap \mathcal{K})$, where the $\min(.)$ of an ordered set is the item that appears first in the order. The $\min(.)$ is taken as 0 for the empty set.

Example II.1. For $N = 6, K = 2$, consider the test $t = (v_3, v_5, v_2, v_1)$. If $\mathcal{K} = \{v_2, v_5\}$, the result will be $y = \min(t \cap \mathcal{K}) = \min((v_5, v_2)) = v_5$.

III. ADAPTIVE SETTING

In the adaptive setting, we are allowed to design tests sequentially using the results of previous tests to design the next test. In this setting, we can recover the defective set $\mathcal{K}$ using at most K tests via the following simple procedure.

Initialise $\hat{\mathcal{K}} \leftarrow \emptyset$, $i \leftarrow 1$ and run the following loop:

1) Run a test which includes the items in $V \backslash \hat{\mathcal{K}}$ in an arbitrary order.
2) If the test returns 0 (meaning it found no defectives), terminate the procedure and return $\hat{\mathcal{K}}$.
3) If the test returns v, then update $\hat{\mathcal{K}} \leftarrow \hat{\mathcal{K}} \cup \{v\}$.
4) Update $i \leftarrow i + 1$. If $i > K$, terminate the procedure and return $\hat{\mathcal{K}}$.

The above procedure finds a new defective item in each test (except perhaps the last one if $|\mathcal{K}| < K$), and thus finds all defective items in at most K tests. This is also the minimum number of tests needed (in the worst case of $|\mathcal{K}| = K$) to find all defectives, as each test enables us to find at most a single defective item. Thus, the above simple procedure is in fact optimal.

IV. NON-ADAPTIVE SETTING

In the non-adaptive setting, we must fix all the tests beforehand. For a given a collection of tests $\mathcal{T} = \{t_1, t_2, ..., t_T\}$ and a (unknown) defective set $\mathcal{K}$, the outputs of the tests are given by $y_i = \min(t_i \cap \mathcal{K})$ for $i = 1, 2, .., T$, which we will collectively represent by the output vector $y := [y_1, y_2, ..., y_T]$. Let $\mathcal{X} := \{S \subset V : |S| \leq K\}$ denote the collection of possible defective sets $\mathcal{K}$ and $\mathcal{Y} := (V \cup \{0\})^T$ denote the set of possible output vectors y. Finally, let $\Omega_{\mathcal{T}} : \mathcal{X} \rightarrow \mathcal{Y}$ represent the function that maps every possible defective set to its output vector.

For given N, K, we will say that a collection of tests $\mathcal{T}$ is a *feasible testing design* if any set of at most K defectives can be identified using the test outputs. Next, we will identify a necessary and sufficient condition for a collection of tests to be feasible. This can be thought of as an analogue of the test matrix *disjunctness* property that plays the same role under standard binary group testing [12, Chapter 1].

Theorem 1. *Given N, K, a testing design $\mathcal{T}$ is feasible if and only if it satisfies the following condition:*

For all subsets $\mathcal{K} \subset V$ with $|\mathcal{K}| = K$, and for every $v \in \mathcal{K}$, there exists $t \in \mathcal{T}$ such that:

$$v = \min(t \cap \mathcal{K}) \iff v \in t, \ v \leq_t x \ \forall x \in t \cap \mathcal{K}. \quad (1)$$

Proof Sketch. In words, the above condition requires that for any possible collection of defectives $\mathcal{K}$ and any item $v \in \mathcal{K}$, there is at least one test t where v appears before every other item in $\mathcal{K}$ and hence v can be identified as defective by t. Formally, the proof follows by noting that $\mathcal{T}$ is a feasible testing design if and only if the map $\Omega_{\mathcal{T}}$ is a one-to-one function, and then proving the necessity and sufficiency of (1) for this. The detailed proof can be found in Appendix A. ∎

A. Reconstruction Algorithm

Given the output of a feasible test design satisfying (1), one needs a reconstruction algorithm to estimate the set of defectives. The following claim suggests one such strategy.

Claim 1. *For a feasible testing design $\mathcal{T} = \{t_1, ..., t_T\}$ and a (unknown) set of defectives $\mathcal{K}$ with $|\mathcal{K}| \leq K$, let $y = [y_1, y_2, ...y_T] = \Omega_{\mathcal{T}}(\mathcal{K})$ be the output vector. Then $\mathcal{K} = \{y_i : i \in [T], y_i \neq 0\}$.*

Proof. Follows directly from Theorem 1, which guarantees that each defective $v \in \mathcal{K}$ is detected by at least one test, i.e., $\exists i \in [T] : y_i = v$. ∎

The above claim provides a natural and simple reconstruction algorithm for cascaded testing. This reconstruction is analogous to the 'smallest satisfying set' reconstruction algorithm, which is popular for standard binary group testing [12, Chapter 2].

V. OPTIMAL TESTING DESIGN

Definition 1 (Optimal testing design). For given N, K, a testing design $\mathcal{T}^*$ is said to be optimal if $\mathcal{T}^*$ is feasible and for every feasible design $\mathcal{T}$, $|\mathcal{T}| \geq |\mathcal{T}^*|$ holds. Define:

$$T(N, K) := \min\{|\mathcal{T}| : \mathcal{T} \text{ is feasible}\}$$

By definition, the size of any optimal design is $T(N, K)$. In the rest of the paper, we derive bounds on the quantity $T(N, K)$.

To assist with this, we now introduce some notation to write the condition in Theorem 1 more compactly. Let $\mathcal{T}$ be a testing design as before. Given any $S \subset V$ with $S \neq \emptyset$ and some $v \in S$, define:

$$f_{\mathcal{T}}(S, v) := |\{t \in \mathcal{T} : v = \min(t \cap S)\}|. \quad (2)$$

This function counts the number of tests in which v appears first among the items in S. Using this we can write the condition in Theorem 1 as follows:

$$\forall \mathcal{K} \subset [N] \text{ s.t. } |\mathcal{K}| = K, \forall v \in \mathcal{K} : f_{\mathcal{T}}(\mathcal{K}, v) \geq 1. \quad (3)$$

We note the following useful property of $f_{\mathcal{T}}$ for any non-empty $S \subset V$:

$$\sum_{v \in S} f_{\mathcal{T}}(S, v) \leq |\mathcal{T}|. \quad (4)$$

This follows from the observation that the sets $\{t \in \mathcal{T} : v = \min(t \cap S)\}$ are pairwise disjoint for $v \in S$.

We will begin by proving a lower bound on $T(N, K)$, for which we first consider testing designs with a particular 'systematic' form.

A. Systematic Form

Definition 2 (Systematic form). A testing design $\mathcal{T} = \{t_1, ..., t_T\}$ is said to be in systematic form if $\forall j \in [T]$, $t_j \neq \emptyset$ and $\min(t_j) \notin t_k \ \forall k \in [T] \setminus \{j\}$.

In words, for a testing design in systematic form, the item appearing first in a test is excluded in other tests. Next, we show that to characterize the minimum number of tests required, one can restrict attention to such testing designs.

Theorem 2. *Given any feasible testing design $\mathcal{T}_1$, we can construct a feasible testing design $\mathcal{T}_2$ in systematic form satisfying $|\mathcal{T}_2| \leq |\mathcal{T}_1|$.*

The proof of the above result can be found in Appendix B. We present an example to illustrate this result.

Example V.1. For $N = 4, K = 3$, consider the feasible design $\mathcal{T}_1 = \{(v_1, v_2, v_3, v_4), (v_3, v_2, v_4, v_1), (v_4, v_2, v_3, v_1)\}$. This can be reduced to the systematic form design $\mathcal{T}_2 = \{(v_1, v_2), (v_3, v_2), (v_4, v_2)\}$, which is also feasible.

The corollary below immediately follows from Theorem 2.

Corollary 1. *For given N, K, let $\mathcal{G}_{N,K}$ be the set of feasible testing designs in systematic form. Then*

$$T(N, K) = \min\{|\mathcal{T}| : \mathcal{T} \in \mathcal{G}_{N,K}\}.$$

We now present an equivalent condition for the feasibility of systematic form designs.

Theorem 3. *Let $\mathcal{T}$ be a testing design in systematic form and let $L := \{\min(t) : t \in \mathcal{T}\}$. Then $\mathcal{T}$ is a feasible testing design if and only if the following holds*

$$\forall S \subset V \setminus L \text{ s.t. } 1 \leq |S| \leq K,$$
$$\forall v \in S : f_{\mathcal{T}}(S, v) \geq K + 1 - |S| \quad (5)$$

Proof. Observe that the $\min(.)$ of distinct tests are distinct in a systematic form design. Define $P := V \setminus L$ for convenience. We first prove the forward implication. Suppose $\mathcal{T}$ is feasible and (5) doesn't hold. Then $\exists S \subset P$ with $1 \leq |S| \leq K$ and $v \in S$ such that $f_{\mathcal{T}}(S, v) \leq K - |S|$. Let $\mathcal{I} := \{t \in \mathcal{T} : v = \min(t \cap S)\}$ and note that $|\mathcal{I}| = f_{\mathcal{T}}(S, v)$. Consider $S_1 := \{\min(t) : t \in \mathcal{I}\} \subset L$, with $|S_1| = |\mathcal{I}| \leq K - |S|$. Also, note that $S_1 \cap S = \emptyset$ since $S_1 \subset L$ and $S \subset P$.

Now consider a set of items $S_2 \supset S_1$ satisfying $S_2 \cap S = \emptyset$ and $|S_2| = K - |S|$. Let $\mathcal{K} := S \cup S_2 \Rightarrow |\mathcal{K}| = K$. Then $\forall t \in \mathcal{I}, \ \min(t) \in S_1 \subset \mathcal{K} \Rightarrow \min(t \cap \mathcal{K}) = \min(t) \neq v$. By definition of $\mathcal{I}, \forall t \in \mathcal{T} \setminus \mathcal{I}, \ \min(t \cap \mathcal{K}) \leq_t \min(t \cap S) <_t v \Rightarrow v \neq \min(t \cap \mathcal{K})$. Thus $\forall t \in \mathcal{T}, \ v \neq \min(t \cap \mathcal{K})$. By Theorem 1, this contradicts our assumption that $\mathcal{T}$ is feasible.

Next, we prove the converse. Suppose (5) is satisfied. We will

check the feasibility of $\mathcal{T}$ using Theorem 1. Consider arbitrary $\mathcal{K} \subset V$ with $|\mathcal{K}| = K$. Let $S_1 = P \cap \mathcal{K}$ and $S_2 = L \cap \mathcal{K}$. Note that $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = \mathcal{K}$.

For each $v \in S_2 \subset L$, $\exists t \in \mathcal{T}, v = \min(t)$ by the definition of L. If $S_1 = \emptyset$, $S_2 = \mathcal{K}$ and we are done.

Suppose $S_1 \neq \emptyset$. For $v \in S_1 \subset P$, we know $f_{\mathcal{T}}(S_1, v) \geq K + 1 - |S_1| = |S_2| + 1$. Let $\mathcal{J} := \{t \in \mathcal{T} : v = \min(t \cap S_1)\}$ and let $S_3 := \{\min(t) : t \in \mathcal{J}\} \subset L$. Then $|S_3| = |\mathcal{J}| = f_{\mathcal{T}}(S_1, v) \geq |S_2| + 1 \Rightarrow S_3 \backslash S_2 \neq \emptyset$. Consider some $u \in S_3 \backslash S_2$ and let $t_u \in \mathcal{J}$ be the test satisfying $u = \min(t_u)$. Since $\mathcal{T}$ is in systematic form, $\forall w \in S_2 : w \notin t_u$. So $\min(t_u \cap \mathcal{K}) = \min(t_u \cap \mathcal{K} \backslash S_2) = \min(t_u \cap S_1) = v$.

Thus the condition in Theorem 1 holds. ∎

B. Lower Bound

With Theorem 3, we are now in a position to prove the following lower bound on the number of tests required by any scheme under cascaded testing.

Theorem 4. *For given N, K, consider $\alpha, \beta \in \mathbb{N}$ with $\alpha + \beta = K + 1$. Then $N \geq \alpha(\beta + 1) - 1 \Rightarrow T(N, K) \geq \alpha\beta$.*

Proof. By Corollary 1, we have $T(N, K) = \min\{|\mathcal{T}| : \mathcal{T} \in \mathcal{G}_{N,K}\}$. Consider any $\mathcal{T} \in \mathcal{G}_{N,K}$. Let $T := |\mathcal{T}|$ and $L := \{\min(t) : t \in \mathcal{T}\}$. Suppose $N \geq \alpha(\beta + 1) - 1$.

Case 1: If $N - T \geq \alpha \Rightarrow |V| - |L| \geq \alpha$. Thus we can choose $S \in V \backslash L$ such that $|S| = \alpha$. Since $1 \leq |S| \leq K$, we can apply Theorem 3 to obtain $f_{\mathcal{T}}(S, v) \geq K + 1 - \alpha = \beta \; \forall v \in S$. Using (4), we get:

$$T \geq \sum_{v \in S} f_{\mathcal{T}}(S, v) \geq |S|\beta = \alpha\beta$$

Case 2: If $N - T \leq \alpha - 1$, we have $T \geq N + 1 - \alpha \geq \alpha(\beta + 1) - \alpha = \alpha\beta$.

$$\Rightarrow |\mathcal{T}| \geq \alpha\beta \; \forall \mathcal{T} \in \mathcal{G}_{N,K} \Rightarrow T(N, K) \geq \alpha\beta. \quad \blacksquare$$

Choosing $\alpha = \lfloor \frac{K+1}{2} \rfloor$, $\beta = \lceil \frac{K+1}{2} \rceil$ in Theorem 4, we get the following $\Omega(K^2)$ lower bound on $T(N, K)$.

Corollary 2. *Suppose $N \geq \lfloor \frac{K+1}{2} \rfloor (\lceil \frac{K+1}{2} \rceil + 1) - 1$, then:*

$$T(N, K) \geq \left\lfloor \frac{K+1}{2} \right\rfloor \left\lceil \frac{K+1}{2} \right\rceil.$$

VI. FEASIBLE TESTING DESIGNS

In this section, we will present upper bounds on the optimal number of tests $T(N, K)$. Firstly, note that since the cascaded testing model provides at least as much information as the standard binary group testing model, any achievable strategy for the latter is also a feasible one for the former. Thus, an upper bound of $O(K^2 \min\{\log_K^2 N, \log N\})$ on $T(N, K)$ follows from [16], [17] which provided explicit constructions for the binary group testing model. Additionally, an upper bound of $O(K^2 \log(N/K))$ can be established via a randomized construction, and is included in Appendix C.

Next, for any constant K and arbitrary N, we present carefully designed explicit testing designs which use order-wise fewer tests than the upped bounds discussed above. We

will first consider $K = 1, 2, 3$ and then describe the general construction. While $K = 1, 2$ are straightforward, the case of $K \geq 3$ is much more challenging. For simplicity, we will henceforth consider the set of items $V = \{1, 2, ..., N\}$ w.l.o.g.

A. $K = 1$

We can meet the condition in Theorem 1 by using a single test with all items included, i.e., $t_1 = (1, 2, ..., N)$. It is easy to see that this test will recover any one defective item.

B. $K = 2$

We can meet the condition in Theorem 1 by using two tests:

$$t_1 = (1, 2, ..., N), \; t_2 = (N, N-1, ..., 1)$$

For any set $\{i, j\} \subset V$ with $i < j$, we have $i <_{t_1} j$ and $j <_{t_2} i$. Again, it is easy to see that the two tests will identify any pair of defective items.

The above constructions are also optimal as they achieve the obvious lower bound of K tests. Thus for $K = 1, 2$ we have $T(N, K)$ independent of the total number of items N. Note that this is in contrast to standard nonadaptive binary group testing where for $K = O(1)$, $\Omega(\log N)$ tests are necessary. This demonstrates the possible reduction in optimal testing design size that the more informative cascaded testing model might allow for.

C. $K = 3$

For $K = 3$, we provide a recursive construction. Suppose we have a feasible testing design $\mathcal{T}_1$ for n items. Additionally, assume that each test in $\mathcal{T}_1$ is a permutation of all n items. We will provide a procedure to construct a feasible design $\mathcal{T}_2$ for n^2 items, with $|\mathcal{T}_2| \leq |\mathcal{T}_1| + 4$.

Procedure $\mathcal{A}$: Let $N = n^2$ and let $V = \{1, ..., N\}$ be the set of N items. Partition V into disjoint sets $A_1, A_2, .., A_n$ where $A_i = \{(i-1)n + 1, (i-1)n + 2, ..., in\}$.

Next, we introduce some notation which will help us describe the construction. Given permutations s_1, s_2 of n items each, we define the permutation $s_3 = s_1 \circ s_2$ of $N = n^2$ items as follows: for each i, arrange the n items of A_i according to s_2 and call the resulting permutation h_i. Next arrange the permutations $h_1, h_2, ..., h_n$ according to the order in s_1 to obtain the overall permutation s_3 on N items. As an example, consider $n = 3$, $N = n^2 = 9$, and $A_i = \{3i - 2, 3i - 1, 3i\}$ for $i = 1, 2, 3$. If $s_1 = (2, 3, 1), s_2 = (1, 3, 2)$, then $h_i = (3i - 2, 3i, 3i - 1)$ and $s_1 \circ s_2 = (h_2, h_3, h_1) = (4, 6, 5, 7, 9, 8, 1, 3, 2)$.

Now suppose $\mathcal{T}_1 = \{t_1, ..., t_{|\mathcal{T}_1|}\}$. Consider the following set of permutations of N items:

$$\mathcal{F} := \{t_i \circ t_i : i \in [|\mathcal{T}_1|]\}.$$

Now, consider two additional permutations of n items given by $g_1 = (1, 2, ..., n)$ and $g_2 = (n, n-1, ..., 1)$. Let

$$\mathcal{H} := \{g_i \circ g_j : i, j \in [2]\}.$$

Note that $|\mathcal{F}| = T$, $|\mathcal{H}| = 4$. Finally, the overall testing design for $N = n^2$ items is given by $\mathcal{T}_2 := \mathcal{F} \cup \mathcal{H}$, and thus $|\mathcal{T}_2| \leq |\mathcal{T}_1| + 4$.

Example VI.1. For $n = 3$ items $\{1, 2, 3\}$, consider the (trivial) feasible testing design $\mathcal{T}_1 = \{(1, 2, 3), (2, 1, 3), (3, 2, 1)\} = \{t_1, t_2, t_3\}$ with $T = 3$ tests. For $N = n^2 = 9$ items $\{1, ..., 9\}$, we use procedure $\mathcal{A}$ to devise a feasible testing design $\mathcal{T}_2$. Set $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5, 6\}$ and $A_3 = \{7, 8, 9\}$. Then we can construct $\mathcal{F}$ containing the following permutations:

$$t_1 \circ t_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9), \ t_2 \circ t_2 = (5, 4, 6, 2, 1, 3, 8, 7, 9),$$
$$t_3 \circ t_3 = (9, 8, 7, 6, 5, 4, 3, 2, 1).$$

We can also construct $\mathcal{H}$ using $g_1 = (1, 2, 3), g_2 = (3, 2, 1)$ as follows

$$g_1 \circ g_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9), \ g_1 \circ g_2 = (3, 2, 1, 6, 5, 4, 9, 8, 7)$$
$$g_2 \circ g_1 = (7, 8, 9, 4, 5, 6, 1, 2, 3), \ g_2 \circ g_2 = (9, 8, 7, 6, 5, 4, 3, 2, 1)$$

This gives $\mathcal{T}_2 = \mathcal{F} \cup \mathcal{H}$ as:

$$\mathcal{T}_2 = \{(1, 2, 3, 4, 5, 6, 7, 8, 9), (9, 8, 7, 6, 5, 4, 3, 2, 1),$$
$$(3, 2, 1, 6, 5, 4, 9, 8, 7), (7, 8, 9, 4, 5, 6, 1, 2, 3)$$
$$(5, 4, 6, 2, 1, 3, 8, 7, 9)\}$$

It can be verified that $\mathcal{T}_2$ above satisfies the condition in Theorem 1, and is thus a feasible design for $N = 9, K = 3$. Claim 2 in the next subsection proves that this is true for any testing design matrix constructed using Procedure $\mathcal{A}$.

Since we are guaranteed feasibility of the constructed testing design, we can now recursively apply Procedure $\mathcal{A}$ starting with $\mathcal{T}_1 = \{(1, 2, 3), (2, 1, 3), (3, 2, 1)\}$, $n = 3$, to get designs for any number of items. Moreover, Procedure $\mathcal{A}$ gives us the following guarantee $\forall n \geq 3$:

$$T(n^2, 3) \leq T(n, 3) + 4. \tag{6}$$

This recursion implies that $T(N, 3) = O(\log \log N)$, which is a significant improvement over standard group testing where at least $\Omega(\log N)$ tests will be required.

D. $K \geq 3$

We now consider a generalisation of Procedure $\mathcal{A}$ to any $K \geq 3$. Suppose we have a feasible testing design $\mathcal{T}_1$ for parameters n, K and another feasible design $\mathcal{G}_1$ for parameters $n, (K - 1)$. Additionally, assume all tests in these designs are permutations of all n items. We will provide a procedure to construct a feasible design $\mathcal{T}_2$ for $N = n^2$ items, with $|\mathcal{T}_2| \leq |\mathcal{T}_1| + |\mathcal{G}_1|^2$.

Generalized Procedure $\mathcal{A}$: Let $N = n^2$ and let $V = \{1, ..., N\}$ be the set of N items. Partition V into disjoint sets $A_1, A_2, .., A_n$ where $A_i = \{(i-1)n+1, (i-1)n+2, ..., in\}$. Now construct $\mathcal{F}, \mathcal{H}$ as follows:

$$\mathcal{F} := \{t \circ t : t \in \mathcal{T}_1\}, \ \mathcal{H} = \{g \circ g' : g, g' \in \mathcal{G}_1\}.$$

Note that $|\mathcal{F}| = |\mathcal{T}_1|$, $|\mathcal{H}| = |\mathcal{G}_1|^2$. Finally, the overall testing design for $N = n^2$ items is given by $\mathcal{T}_2 := \mathcal{F} \cup \mathcal{H}$, and thus $|\mathcal{T}_2| \leq |\mathcal{T}_1| + |\mathcal{G}_1|^2$.

Note that for $K = 3$, we can use $\mathcal{G}_1 = \{(1, 2, ..., n), (n, n-1, ..., 1)\}$ which is feasible for parameters $n, (K-1) = 2$. This reduces the procedure to Procedure $\mathcal{A}$ for $K = 3$.

The following claim establishes the feasibility of the design obtained from the generalized Procedure $\mathcal{A}$.

Claim 2. *The testing design $\mathcal{T}_2$ obtained from the generalized Procedure $\mathcal{A}$ is feasible for parameters N, K.*

Proof Sketch. We check that $\mathcal{T}_2$ satisfies the condition for feasibility in Theorem 1 for parameters N, K. We do this by considering 3 cases for $\mathcal{K} \subset V$:
  1) When all items of $\mathcal{K}$ fall into a single partition A_i, the feasibility of $\mathcal{T}_1$ enables us to find a permutation in $\mathcal{F}$ for each $v \in \mathcal{K}$, where v is the first defective.
  2) When all items of $\mathcal{K}$ fall into K separate partitions, the idea above still applies and can be used to find suitable permutations in $\mathcal{F}$.
  3) When neither of the above cases happens, there will be $L \leq K - 1$ partitions into which items of $\mathcal{K}$ fall, and each partition has at most $K - 1$ items. Then for each $v \in \mathcal{K}$, the feasibility of $\mathcal{G}_1$ for $n, K - 1$ allows us to find a permutation in $\mathcal{H}$, in which v is the first defective.
The detailed proof can be found in Appendix D. ∎

Given feasible designs for parameters $N, K-1$ for arbitrary N, one can use this procedure to recursively construct feasible designs for N, K for arbitrary N. Since we have feasible designs for $K = 2$, we can recursively obtain feasible designs for arbitrary K. This procedure also guarantees that $\forall n \geq K$:

$$T(n^2, K) \leq T(n, K) + T(n, K - 1)^2. \tag{7}$$

This allows us to prove the following result:

Claim 3. *For a fixed $K \geq 3$, we have $T(N, K) = O((\log \log N)^{a_K})$, where $a_K = 2^{(K-2)} - 1$.*

The proof can be found in Appendix E. This result implies that $T(N, K)$ grows at most as a polynomial in $\log \log N$ for a fixed K. This is a significant improvement over standard group testing, where the size of designs is $\Omega(\log N)$.

VII. DISCUSSION

While we were able to present bounds on the optimal size $T(N, K)$ for cascaded group testing, several pertinent questions remain open. Firstly, our lower bound of $\Omega(K^2)$ from Corollary 2 does not have any dependence on the total number of items N, as opposed to the general upper bounds we have for $K \geq 3$. Deriving a stronger lower bound to close this gap, starting with the specific case of $K = 3$, is an immediate direction that we are pursuing. Secondly, while we were able to demonstrate explicit (near)-optimal constructions for fixed values of K, the general case is open. This will also help provide a characterization of the advantage that the additional information in a cascaded test provides over a standard binary OR test. Beyond the zero-error noiseless setting studied here, considering recovery with small error probability and various forms of noise in the test output are interesting directions to pursue.

REFERENCES

[1] R. Dorfman, "The detection of defective members of large populations," *The Annals of mathematical statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[2] J. Wolf, "Born again group testing: Multiaccess communications," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 185–191, 1985.

[3] A. Sharma and C. R. Murthy, "Group testing-based spectrum hole search for cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3794–3805, 2014.

[4] H. A. Inan, S. Ahn, P. Kairouz, and A. Ozgur, "A group testing approach to random access for short-packet communication," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 96–100.

[5] F. K.-m. Hwang and D.-z. Du, *Pooling designs and nonadaptive group testing: important tools for DNA sequencing*. World Scientific, 2006, vol. 18.

[6] J. Engels, B. Coleman, and A. Shrivastava, "Practical near neighbor search via group testing," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9950–9962, 2021.

[7] J. Luo and D. Guo, "Neighbor discovery in wireless ad hoc networks based on group testing," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 791–797.

[8] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, "Graph-constrained group testing," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 248–262, 2012.

[9] M. Mukamoto, T. Matsuda, S. Hara, K. Takizawa, F. Ono, and R. Miura, "Adaptive boolean network tomography for link failure detection," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 646–651.

[10] D. Du, F. K. Hwang, and F. Hwang, *Combinatorial group testing and its applications*. World Scientific, 2000, vol. 12.

[11] M. Malyutov, "Search for sparse active inputs: A review," *Information Theory, Combinatorics, and Search Theory: In Memory of Rudolf Ahlswede*, pp. 609–647, 2013.

[12] M. Aldridge, O. Johnson, J. Scarlett *et al.*, "Group testing: an information theory perspective," *Foundations and Trends® in Communications and Information Theory*, vol. 15, no. 3-4, pp. 196–392, 2019.

[13] F. K. Hwang, "A method for detecting all defective members in a population by group testing," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 605–608, 1972.

[14] A. G. D'yachkov and V. V. Rykov, "Bounds on the length of disjunctive codes," *Problemy Peredachi Informatsii*, vol. 18, no. 3, pp. 7–13, 1982.

[15] Z. Füredi, "Onr-cover-free families," *Journal of Combinatorial Theory, Series A*, vol. 73, no. 1, pp. 172–173, 1996.

[16] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 363–377, 1964.

[17] E. Porat and A. Rothschild, "Explicit non-adaptive combinatorial group testing schemes," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2008, pp. 748–759.

[18] M. Cheraghchi, "Improved constructions for non-adaptive threshold group testing," in *Automata, Languages and Programming: 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I 37*. Springer, 2010, pp. 552–564.

[19] H.-B. Chen and H.-L. Fu, "Nonadaptive algorithms for threshold group testing," *Discrete Applied Mathematics*, vol. 157, no. 7, pp. 1581–1585, 2009.

[20] M. Soleymani, H. Mahdavifar, and T. Javidi, "Non-adaptive quantitative group testing via plotkin-type constructions," in *2023 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2023, pp. 1854–1859.

[21] T. V. Bui and J. Scarlett, "Concomitant group testing," *arXiv preprint arXiv:2309.04221*, 2023.

[22] B. Kveton, C. Szepesvari, Z. Wen, and A. Ashkan, "Cascading bandits: Learning to rank in the cascade model," in *International conference on machine learning*. PMLR, 2015, pp. 767–776.

[23] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari, "Combinatorial cascading bandits," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[24] C. Gan, R. Zhou, J. Yang, and C. Shen, "Cost-aware cascading bandits," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3692–3706, 2020.

Proof. It is easy to see that $\mathcal{T}$ is a feasible testing design if and only if $\Omega_{\mathcal{T}}$ is a one-to-one function. We will now argue that (1) is necessary and sufficient for $\Omega_{\mathcal{T}}$ to be one-to-one.

First, we prove the forward implication. Suppose $\Omega_{\mathcal{T}}$ is one-one, but condition (1) does not hold. Then $\exists \mathcal{K} \subset V$ with $|\mathcal{K}| = K$ and $\exists v \in \mathcal{K}$, such that $\forall t \in \mathcal{T}$ one of the following holds: a) $v \notin t$, or b) $v \in t$ and $\exists x \in \mathcal{K} \cap t$ satisfying $x <_t v$. In case a), we have $\min(t \cap \mathcal{K}) = \min(t \cap \mathcal{K} \backslash \{v\})$.
In case b), we must have $\min(t \cap \mathcal{K} \backslash \{v\}) \leq_t x <_t v$. Thus, $\min(t \cap \mathcal{K} \backslash \{v\}) = \min(t \cap \mathcal{K})$, since the first item of a test is not changed upon adding an item after it.
Thus $\forall t \in \mathcal{T}$, $\min(t \cap \mathcal{K}) = \min(t \cap \mathcal{K} \backslash \{v\}) \Rightarrow \Omega_{\mathcal{T}}(\mathcal{K}) = \Omega_{\mathcal{T}}(\mathcal{K} \backslash \{v\})$. This is a contradiction since $\Omega_{\mathcal{T}}$ was assumed to be one-one.

Next, we prove the converse, i.e., condition (1) implies $\Omega_{\mathcal{T}}$ is one-one. Consider any $S_1, S_2 \in \mathcal{X}$ with $S_1 \neq S_2$. Assume w.l.o.g. that $S_1 \backslash S_2 \neq \emptyset$. So $\exists x_1 \in S_1$ such that $x_1 \notin S_2$. Consider some $\mathcal{K} \supset S_1$ with $|\mathcal{K}| = K$. By (1), we have that $\exists t \in \mathcal{T}$ such that $x_1 \in t$ and $x_1 <_t x \ \forall \ x \in \mathcal{K} \cap t \Rightarrow \min(t \cap S_1) = x_1$. Also, $\min(t \cap S_2) \neq x_1$ since $x_1 \notin S_2$. Thus, $\min(t \cap S_1) \neq \min(t \cap S_2)$

$$\Rightarrow \Omega_{\mathcal{T}}(S_1) \neq \Omega_{\mathcal{T}}(S_2).$$

Since the above argument holds for any distinct $S_1, S_2 \in \mathcal{X}$, we have that (1) implies $\Omega_{\mathcal{T}}$ is one-one. ∎

Before we prove Theorem 2, we prove an intermediate result.

Lemma 1. *Suppose we have a feasible testing design $\mathcal{T} = \{t_1, ..., t_T\}$. Given $l \in [T]$ s.t. $t_l \neq \emptyset$, define $\hat{t}_i = t_i \backslash \{min(t_l)\}$ for $i \in [T] \backslash \{l\}$ and $\hat{t}_l = t_l$. Then the design $\hat{\mathcal{T}} = \{\hat{t}_1, ..., \hat{t}_T\}$ is feasible.*

Proof. Let $u := \min(t_l)$. We check if $\hat{\mathcal{T}}$ satisfies (3) for cases when $u \in \mathcal{K}$ and when $u \notin \mathcal{K}$.
If $u \notin \mathcal{K}$, then $\min(t \cap \mathcal{K}) = \min(t \cap \mathcal{K} \backslash \{u\}) \ \forall t \in \mathcal{T}$, which implies that $\forall v \in \mathcal{K}$, $f_{\hat{\mathcal{T}}}(\mathcal{K}, v) = f_{\mathcal{T}}(\mathcal{K}, v) \geq 1$.
If $u \in \mathcal{K}$, consider $v \in \mathcal{K}$. If $v = u$, we have $f_{\hat{\mathcal{T}}}(\mathcal{K}, v) \geq 1$ since $u = \min(\hat{t}_l \cap \mathcal{K})$. If $v \neq u$, consider $j \in [T]$ such that $v = \min(t_j \cap \mathcal{K})$. Then $j \neq l$ and we have $\min(\hat{t}_j \cap \mathcal{K}) = \min(t_j \cap \mathcal{K} \backslash \{u\}) = v \Rightarrow f_{\hat{\mathcal{T}}}(\mathcal{K}, v) \geq 1$.
This covers all cases and thus $\hat{\mathcal{T}}$ is feasible. ∎

With this lemma, we are ready for the proof of Theorem 2.

Proof of Theorem 2. : First, we make a simple observation: Removing an empty test from a feasible design does not change its feasibility.
Applying this idea, we first remove any empty test from $\mathcal{T}_1$ to obtain the feasible design $\{h_1, h_2, ..., h_T\}$. Now we apply a procedure to construct a systematic form design.
Initialise $\mathcal{E} \leftarrow \emptyset$, $i \leftarrow 1$ and run the following loop:

1) If $h_i = \emptyset$: Update $\mathcal{E} \leftarrow \mathcal{E} \cup \{i\}$. Update $i \leftarrow i + 1$ and start the loop again.
2) $\forall j \in [T] \backslash \{i\}$, update $h_j \leftarrow h_j \backslash \{\min(h_i)\}$.
3) Update $i \leftarrow i + 1$. If $i > T$, terminate the procedure.

After the procedure ends, set $\mathcal{T}_2 = \{h_j : j \in [T] \backslash \mathcal{E}\}$.
The design $\{h_1, ..., h_T\}$ obtained after each step of the above procedure is feasible by application of Lemma 1. During the ith loop iteration, whenever $h_i \neq \emptyset$, the update in step 2 results in $\min(h_i) \notin h_j \ \forall j \in [T] \backslash \{i\}$. Since each iteration can only result in the removal of items from tests, $\min(h_i)$ stays excluded from all other tests till the end. Thus in a later iteration i', $\min(h_i)$ is not removed from h_i, since $\min(h_{i'}) \neq \min(h_i)$. This guarantees that after the procedure terminates, all empty tests are marked in $\mathcal{E}$, and the remaining tests lead to a systematic form design. ∎

We now present an upper bound on $T(N, K)$, which is based on a randomized construction.

Claim 4. $\forall N, K$ *with* $K \geq 2$,

$$T(N, K) \leq \left\lfloor \log_{\frac{K}{K-1}} \left(K \binom{N}{K} \right) \right\rfloor + 1 \leq \left\lceil K \log \left(K \binom{N}{K} \right) \right\rceil$$

Proof. We use a probabilistic argument to prove this result for any given N, K. Choose tests $t_1, t_2, ..., t_T$ independently and uniformly at random from the set of permutations of V. We look at the probability of the event E that $\mathcal{T} = \{t_1, ..., t_T\}$ is not feasible. Using (3) we obtain

$$E = \bigcup_{\substack{v \in \mathcal{K} \subset V, \\ |\mathcal{K}| = K}} \{f_{\mathcal{T}}(\mathcal{K}, v) = 0\}. \tag{8}$$

From (2), we have that for any $\mathcal{K}, v$ in the union,

$$\{f_{\mathcal{T}}(\mathcal{K}, v) = 0\} = \bigcap_{j \in [T]} A_j^c, \ A_j = \{v \leq_{t_j} x, \ \forall x \in \mathcal{K}\}.$$

Since each test t_j is chosen uniformly at random, any element of $\mathcal{K}$ is equally likely to appear first among elements of $\mathcal{K}$. Thus, $\Pr[A_j] = 1/K$. From the independence of t_j's, we have

$$\Pr[f_{\mathcal{T}}(\mathcal{K}, v) = 0] = \left(1 - \frac{1}{K}\right)^T$$

Applying the union bound to (8) gives

$$\Pr[E] \leq K \binom{N}{K} \left(1 - \frac{1}{K}\right)^T$$

If $T > \log_{\frac{K}{K-1}} \left(K \binom{N}{K} \right)$, we obtain $P[E] < 1$. Thus, there must exist some feasible testing design of size $T = \left\lfloor \log_{\frac{K}{K-1}} \left(K \binom{N}{K} \right) \right\rfloor + 1$, which can be easily upper bounded by $\left\lceil K \log \left(K \binom{N}{K} \right) \right\rceil$. ∎

APPENDIX D
PROOF OF CLAIM 2

Proof. We verify that $\mathcal{T}_2$ satisfies the condition in Theorem 1 for parameters N, K. Consider some $\mathcal{K} \subset V$, with $|\mathcal{K}| = K$. Then, we have the following cases:

1) Case 1: $\exists i \in [n] : \mathcal{K} \subset A_i$, i.e., all K items are in the same partition. By feasibility of $\mathcal{T}_1$, for any $v \in \mathcal{K}$, $\exists t \in \mathcal{T}_1 : v = \min(h_i \cap \mathcal{K})$, where h_i is the items of A_i arranged according to t. Let $f := t \circ t \in \mathcal{T}_2$. We have $h_i = (t \circ t) \cap A_i = f \cap A_i$, which gives:
$$\Rightarrow v = \min(f \cap A_i \cap \mathcal{K}) = \min(f \cap \mathcal{K}).$$

2) Case 2: $|\mathcal{K} \cap A_i| \leq 1 \; \forall i \in [n]$. This means there are distinct $i_1, i_2, ..., i_K \in [n]$, and $v_1, v_2, ..., v_K \in V$ satisfying $v_j \in A_{i_j} \forall j \in [K]$ and $\mathcal{K} = \{v_1, v_2, ..., v_K\}$. Let $\mathcal{I} = \{i_1, i_2, ..., i_K\}$. Since $\mathcal{T}_1$ is feasible, $\forall j \in [K]$, $\exists t \in \mathcal{T}_1 : i_j = \min(t \cap \mathcal{I})$. For $k \in [n]$, let h_k denote the items of A_k arranged according to t. Let $f = t \circ t \in \mathcal{T}_2$. Since $\mathcal{K} \subset \cup_{i \in \mathcal{I}} A_i$, we have
$$\min(f \cap \mathcal{K}) = \min(f \cap (\cup_{i \in \mathcal{I}} A_i) \cap \mathcal{K})$$
$$= \min(h_{i_j} \cap \mathcal{K}) = v_j$$

The second line above follows from the fact that i_j appears before other elements of $\mathcal{I}$ in t, which means h_{i_j} appears before h_i in f for all other $i \in \mathcal{I}$. This combined with the fact that h_{i_j} includes all elements of A_{i_j} implies $h_{i_j} \cap \mathcal{K} = \{v_j\}$, giving us the result.

3) Case 3: $|\mathcal{K} \cap A_i| < K \; \forall i \in [n]$ and $\exists i \in [n] : |\mathcal{K} \cap A_i| > 1$. This means there are distinct $A_{i_1}, A_{i_2}, ..., A_{i_L}$ with $1 < L \leq K-1$, such that $0 < |\mathcal{K} \cap A_{i_k}| \leq K-1 \; \forall k \in [L]$ and $\mathcal{K} \subset \cup_{k \in [L]} A_{i_k}$. For a given $v \in \mathcal{K}$, let A_{i_j} be the partition that contains v. Let $\mathcal{K}_j = \mathcal{K} \cap A_{i_j}$, $\mathcal{I} = \{i_k : k \in [L]\}$. Note that $0 < |\mathcal{K}_j| \leq K-1$. By feasibility of $\mathcal{G}_1$ for parameters $n, K-1$, $\exists g, g' \in \mathcal{G}_1 : i_j = \min(g \cap \mathcal{I})$ and $v = \min(h_{i_j} \cap \mathcal{K}_j)$, where h_{i_j} denotes the items of A_{i_j} arranged according to g'. Let $f = g \circ g' \in \mathcal{T}_2$. Then, by the same reasoning as in Case 2, we have
$$\min(f \cap \mathcal{K}) = \min(f \cap (\cup_{i \in \mathcal{I}} A_i) \cap \mathcal{K})$$
$$= \min(h_{i_j} \cap \mathcal{K}) = v$$

In all cases, the condition in Theorem 1 holds, which proves the feasibility of $\mathcal{T}_2$. ∎

APPENDIX E
PROOF OF CLAIM 3

Proof. We prove this by induction on K. First, we show that the claim holds for $K = 3$. Firstly, $a_3 = 2^1 - 1 = 1$. Using equation (6) and the fact that $T(3, 3) = 3$, we get for $r \in \mathbb{N}$ that $T(3^{2^r}, 3) \leq 4r + 3$.

For arbitrary $N \geq 3$, let $r := \lceil \log_2 \log_3(N) \rceil$, then $3^{2^r} \geq N$. By monotone property of $T(N, K)$, we have:
$$T(N, 3) \leq T(3^{2^r}, 3) \leq 4r + 3 = O(\log \log N)$$

Thus, the claim holds for $K = 3$.

Now, suppose the claim holds for some $K \geq 3$. We will show that it also holds for $K + 1$. By assumption, there exists $N_1 \in \mathbb{N}, C_1 \in \mathbb{R}$, s.t., $\forall N \geq N_1$:
$$T(N, K) \leq f(N) := C_1 (\log \log N)^{a_K}$$

We assume $N_1 \geq K+1$ (w.l.o.g.). For arbitrary $r > 0$ consider $N = N_1^{2^r}$. Applying equation (7) repeatedly, we have:
$$T(N, K+1) \leq T(N_1, K+1) + \sum_{l=0}^{r-1} T(N_1^{2^l}, K)^2$$
$$\leq N_1 + r T(N, K)^2 \leq N_1 + r f(N)^2$$

We have used the fact that $T(N, K) \leq N$ and that $T(N, K)$ is monotone in its parameters. Now $r = \log_2 \log_{N_1}(N) \leq C_2 \log \log N$ for a constant $C_2 > 0$ (assuming N sufficiently large). Thus, we have
$$T(N, K+1) \leq N_1 + C_1 C_2 (\log \log N)^{2a_K + 1}$$
$$= N_1 + C_1 C_2 (\log \log N)^{2^{K-1} - 1} \leq C_3 (\log \log N)^{a_{K+1}},$$

for some $C_3 > 0$, assuming $N \geq N_2$ for a suitable $N_2 \in \mathbb{N}$. For arbitrary $N \geq N_2$, let $r' := \lceil \log_2 \log_{N_1}(N) \rceil$, then $N^2 \geq N' := N_1^{2^{r'}} \geq N$. By monotone property of $T(N, K)$, we have $\forall N \geq N_2$:
$$T(N, K+1) \leq T(N', K+1) \leq C_3 (\log \log N')^{a_{K+1}}$$
$$\leq C_3 (\log \log (N^2))^{a_{K+1}} = O((\log \log N)^{a_{K+1}})$$

This completes the proof via induction. ∎