[1]

THE GRAHAM-KNOWLTON PROBLEM REVISITED

Navin Goyal[2]
Sachin Lodha[3]
S. Muthukrishnan[4]

Abstract

In late 60's, Graham and Knowlton introduced the WIP (wire identification problem) that affected electricians: match the wires in the ceiling to those in the basement while making the fewest trips. We revisit this problem and study its variants and generalizations; we provide a combinatorial characterization of the solution(s) in terms of an associated hypergraph and obtain nearly tight bounds on the minimum number of trips, thereby pleasing electricians.
Keywords: Combinatorial structures, combinatorial algorithms, group testing.

1. *Introduction*

For FUN, one must go back to 60's. Graham and Knowlton [2, 3] introduced the WIP which we study in this paper in its variations and generalizations.

Wire Identification Problem (WIP). Suppose that there is a cable consisting of n insulated wires that goes from the basement of a building to its top floor. All the wires look alike. The wires get jumbled on the way. Therefore one does not know how the n terminals at the bottom are *matched* with the n terminals at the top. An electrician needs to determine this matching. He can electrically connect disjoint sets of wires at the basement, and in one trip to the top floor, he can determine the groups of wires that are connected by testing circuit continuity. If two groups of equal sizes are connected at the bottom, he would determine the two groups of wires at the top floor but not the precise group to which the wires belong. Thus each trip can provide partial information about the matching between the terminals at the two ends. He can make multiple trips, in each trip connecting different subsets at the basement. The *wire identification problem* (WIP) is to find the minimum number of trips needed to determine the matching, as well as, to design an algorithm that finds the minimum-trip solution. □

Note that each trip consists of connecting groups of wires at the bottom, going to the top, and checking the connections. We will only be concerned with the number of trips as the measure of cost; the time it takes for the electrician to determine the matching after he has made the requisite number of trips is not considered.

Two observations are immediate. First, it is impossible to determine the matching for $n = 2$, but there is an algorithm for all $n > 2$. Second, this problem is in the large class of what are called *combinatorial group testing problems* where each test is a "group" and the goal is to find a distinguished "configuration" using the group tests [1]. Within this rather general class of problems, there is tremendous variation depending on the nature of the tests allowed. In the combinatorial group testing parlance, the observation is that it suffices to only consider *non-adaptive* procedures *i.e.* procedures in which the connections made at a particular step do not depend on the outcome of prior connections and trips.

This problem is worth pondering about. In our experience with posing this problem as a puzzle, it does not take long for theoretical computer scientists to design a testing algorithms that uses $O(\log n)$ trips by doubling the number of wires that get matched in each trip. Similarly, some note that groups of different sizes give more information than groups of identical sizes. Then a $O(\log\log n)$ round solution ensues by testing groups of sizes $2, 3, 4, ..., O(\sqrt{n})$. This solution is a nice example of the $T(n) = 1+T(O(\sqrt{n})$ recurrence that is hard to motivate in a basic algorithms course, but natural nevertheless. The minimum number of trips needed is 2. Given this solution that readers would have by

[1] We dedicate this paper to the loving memory of Sachin's father, Premsukh Lodha, who passed away on 28$^{\text{th}}$ March, 2004. Being an electrical engineer himself, he showed great interest in this problem and its solution.

now converged to (what follows will be really engaging if you, the readers, solve the puzzle first), the following questions are intriguing:

1. What is the minimum number of trips needed to solve the WIP if we restrict the size of the groups to at most 2?
2. What is the minimum number of trips needed to solve the WIP if we restrict the groups of wires that get connected at the bottom to be *hierarchical*, ie., if groups g_1 and g_2 of the wires get connected in two different trips, we have either $g_1 \cap g_2 = \phi$ or $g_i \subset g_j$ for $i \neq j$ and $i, j \in \{1, 2\}$.

The former is intriguing because readers fairly quickly convince themselves that groups that get tested have to be of different sizes in order to get maximum information, and with the bound on the group size, one expects the number of trips needed to be lot larger than in the basic puzzle. The latter is intriguing because the solution to the basic puzzle with 2 trips hinted above has the "shifting" or "overlapping" property where in the second trip, each group comprises wires drawn from two or more of the groups in the previous trip. Hence, the groups tested are not hierarchical. Again, one senses that such a property is crucial for the solution. In what ensues, we will provide solutions to both these variations and others.

The WIP may also remind the readers of the puzzle about 3 light bulbs in a room with three matching switches in another room; you start in the switch room and must determine the matching using only one visit to the bulb room. There the solution is searing, and one may not want to try a similar solution in the WIP!

We will now formally discuss the previous work and present the problems we address in this paper.

1.1. *Previous Work*

In their original work, Graham and Knowlton [2] gave a solution involving 2 trips using a certain type of partitions (called Knowlton-Graham (KG) partitions by Knuth [5]).

Definition Partitions $\mathcal{A} = \{A_1, A_2, \ldots, A_p\}$ and $\mathcal{B} = \{B_1, B_2, \ldots, B_q\}$ of $[n] = \{1, 2, \ldots, n\}$ are called *KG partitions* of $[n]$ if at most one element appears in an $\mathcal{A}$ set of cardinality j and in a $\mathcal{B}$ set of cardinality k, for each j and k. □

It is easy to see that given KG partitions $\mathcal{A}$ and $\mathcal{B}$ of $[n]$, one can partition $\mathcal{A}$ in the first trip and $\mathcal{B}$ in the second trip; one can then readily identify the matching using the coordinates (j, k). Thus only 2 trips suffice.

EXAMPLE 1 *Let $m > 1$ and $n = \binom{m+1}{2}$. Consider partitions $\mathcal{A} = \{A_1, \ldots, A_m\}$ and $\mathcal{B} = \{B_1, \ldots, B_m\}$ of $[n]$, where $A_i = \{\binom{i}{2} + j \mid 1 \leq j \leq i\}$ and $B_j = \{j + \binom{i}{2} \mid j \leq i \leq m\}$ for $1 \leq i, j \leq m$. It is an easy verification that $\mathcal{A}$ and $\mathcal{B}$ form KG partitions since*

$$A_i \cap B_j = \begin{cases} \{\binom{i}{2} + j\} & \text{if } 1 \leq j \leq i \leq m, \\ \emptyset & \text{otherwise.} \end{cases}$$

Graham [3] proved that KG partitions exist for all n except $2, 5$, and 9. Knuth [5] constructed them using 0-1 matrices.

1.2. *Our Variants*

It is easy to see that KG partitions must have some set of size at least $\Omega(\sqrt{n})$. The question then arises that whether it is necessary to have sets of large sizes in order to solve the WIP. In particular, what happens if we only allow sets of size at most 2? One might be confronted with such a situation when the testing equipment is limited. This is the *first variant* we study. Surprisingly it again turns out that 2 trips are always enough (except for $n = 2$) as we will show soon. Moreover, the solutions have very simple and uniform structure independent of n unlike KG partitions (which are not always trivial to find, and do not exist for $n = 2, 5, 9$). The *second variant* we study is a generalization of the Graham-Knowlton WIP problem: now we have WIP(n, k, x) which is the minimum number of trips needed to solve the WIP problem on n wires where each trip involves testing at most k groups, and each group has cardinality at most x. This is quite a realistic variant because the number of groups that can be tested is limited by the number of parallel circuits that can be tested in each trip. The Graham-Knowlton problem is about WIP(n,n,n) and the first variant above is about WIP$(n,n,2)$. We provide nearly tight bounds for the general quantity WIP(n,k,x). The *third variant* we study is the *hierarchical* testing problem defined earlier where the groups tested form a hierarchy. That is, once groups are formed for testing, for each subsequent trip, we can *only* disconnect some of the wires from the groups (ie, decimate groups), but not reconnect the wires to form new groups. In order words, each group in a testing round is a subset of wires from some other group in the previous testing round. This is a *top-down* view. Here, we do not have to disconnect and reconnect wires of a group. In fact, once the groups are set up for the first trip, we only perform disconnect operations. This variant is very natural: if this was based on soldering connections, then burning out connections is easier than resoldering; in chip manufacturing with FPGAs, burning out connections is more scalable than refabricating connections. There is one technical problem with this variant that it is possible that at some step of the above procedure we get a group of size 2, for which we cannot determine the matching under the hierarchical restriction. One way to get around this is by being satisfied by groups of size 2. So the problem we need to solve under the hierarchical restriction is to divide the set of terminals at the top into groups of size at most 2 so that we know for each group which terminals at the bottom are correspond to it. We study WIP(n,n,n), the original Graham-Knowlton problem under the restriction that the groups be hierarchical. Note that KG-partitions are *not* hierarchical, so one needs a different approach. In this case, we show that $\Theta(\lg^* n)$ (lg denotes logarithm to base 2) is the number of rounds needed, which shows the separation between the hierarchical and the non-hierarchical cases.

Why did we study these variations and generalizations? We believe they are the natural questions that algorithmicists ask when given a puzzle (the original WIP problem). Namely, we study the effect of limiting various resources (number of groups that can be tested in parallel in one trip, number of wires that can be tested in a group, making groups that do not involve *both* disconnections and reconnections, etc.) When we ask such natural questions by limiting resources, the original puzzle that has a discrete math flavor evolves to have an algorithmic flavor. The $\Theta(\lg^* n)$ upper bound has a distinctly algorithmic flavor as does the lower bound in terms of the decision tree. This is how algorithmicists have fun.[5]

In addition, our solution to all the problems is based on a characterization of the solutions to the WIP using the automorphism structure of a hypergraph associated with the problem. This is an abstract way to think about WIP, and is useful in proving upper and lower bounds. This is another way algorithmicists have fun: making a coherent formulation and characterization of a problem.

The rest of the paper is organized as follows. In Section 2 we present our characterization of solutions to the WIP using hypergraph automorphisms. Using this characterization we study WIP(n,k,x). In Section 3 we solve the WIP with sets of size 2 ($x=2$) and extend it to general set sizes (arbitrary x) in Section 4. Finally, we study the hierarchical version in Section 5. We present other variations of interest in Section 6.

2. *Characterization*

We start with a general characterization of when a given testing procedure can identify the matching.

Consider WIP(n,k,x) and let $\mathcal{P}$ be a procedure that solves the problem. $\mathcal{P}$ tells us which wires to connect at the bottom in each trip. These connections can be thought of as labelled hyperedges in a hypergraph on n vertices, namely, $b_1, b_2, \ldots, b_n$, where these vertices correspond to the terminals at the bottom, and label on an hyperedge is the number of the trip in which the subset of vertices is connected. We call this hypergraph the *connection hypergraph*, and denote it by $CG^{\mathcal{P}}$. It completely specifies the testing procedure.

What the electrician observes on the top is completely specified by another hypergraph, which we call the *test hypergraph*, and denote it by $TG^{\mathcal{P}}$. Test hypergraph has the terminals on the top level as vertices, namely, $t_1, t_2, \ldots, t_n$, and the terminals are united by an hyperedge with label i iff the subset of terminals is connected in the ith trip. A hyperedge may get more than one label because it may exist in more than one trip. We say that two test hypergraphs TG_1 and TG_2 are *isomorphic* if there is a bijection between their vertex sets that maps the hyperedges in TG_1 to the hyperedges in TG_2 with the same label set.

THEOREM 1 *A procedure $\mathcal{P}$ can determine the matching of the terminals at the two ends if and only if the automorphism group of its connection hypergraph $CG^{\mathcal{P}}$ is trivial (*i.e.*, it only contains the identity).*

[5] We should really patent these things *a la* [2].–GLM

Proof: The goal of the electrician is to label the vertices $t_1, \ldots t_n$ of the test hypergraph $TG^{\mathcal{P}}$ with $b_1, \ldots, b_n$ so that t_i gets the label $b(t_i)$, its matching terminal at the bottom. The information available to him is the connection hypergraph $CG^{\mathcal{P}}$ at the bottom, and the test hypergraph $TG^{\mathcal{P}}$ at the top.

Clearly $CG^{\mathcal{P}}$ and $TG^{\mathcal{P}}$ are isomorphic. Therefore each automorphism of $CG^{\mathcal{P}}$ defines a labelling of the vertices of $TG^{\mathcal{P}}$ by the b_i's. Each such labelling is consistent with the tests. So the procedure succeeds iff there is exactly one automorphism; in other words, when the automorphism group is trivial. □

Theorem 1 leads us to some useful observations.

For a procedure $\mathcal{P}$ that solves the WIP, let $t^{\mathcal{P}}$ denote its number of trips. Let $E^{\mathcal{P}}$ denote the total number of hyperedges in $CG^{\mathcal{P}}$. Let $d_j^{\mathcal{P}}$ denote the number of vertices that have degree j in $CG^{\mathcal{P}}$. A vertex of degree j appears in some j number of trips in protocol $\mathcal{P}$ since it can not appear in two different hyperedges on the same trip. The size of the automorphism group of $TG^{\mathcal{P}}$ is at least $d_0^{\mathcal{P}}!$. Thus, in the light of theorem 1, $d_0^{\mathcal{P}} \in \{0, 1\}$. Also, there are at most $kt^{\mathcal{P}}$ hyperedges. No two vertices of degree 1 in $CG^{\mathcal{P}}$ belong to the same hyperedge (otherwise they give rise to at least 2 different automorphisms for $TG^{\mathcal{P}}$ violating theorem 1). Thus it follows that $kt^{\mathcal{P}} \geq E^{\mathcal{P}} \geq d_1^{\mathcal{P}}$. These observations will be useful in section 4.

3. *Only Sets of Size* 2 *Allowed*

In this section, we study the "only pairing allowed" variant, *i.e.* $x = 2$ case. First we assume there is no restriction on k, *i.e.* the number of pairs that can be formed per trip.

LEMMA 2 *There exists an optimal* 2 *trip solution for WIP given any* $n > 2$, $k < \infty$, *and* $x = 2$.

Proof: Observe that just one trip is insufficient to solve WIP, so minimum 2 trips are always needed. Now suppose $n > 2$ is odd. Consider following pairings for trip 1 and trip 2 respectively:

$$(b_1, b_2), (b_3, b_4), \ldots, (b_{n-2}, b_{n-1}), b_n$$

and

$$b_1, (b_2, b_3), (b_4, b_5), \ldots, (b_{n-1}, b_n).$$

The connection graph is a path $(b_1, b_2, \ldots, b_n)$ with alternate (and equal number of) edges labelled 1 and 2. Clearly it has the trivial automorphism group. Also, it is trivial for the electrician to determine the matching after the two trips. For even $n > 2$, we use above construction for the case of $n - 1$, leaving out b_n altogether. □

We now consider bounds on k and study WIP$(n, k, 2)$. First observe that: *Given any two natural numbers d and v $(> d)$ satisfying $vd \equiv 0 \mod 2$, there exists a d-regular graph on v vertices.* Proof is left as an exercise to the reader.

THEOREM 3 *Suppose we are given integers* $n, k > 0$ *that satisfy* $n > 3k^2/2 + 1$. *If*

$$\left\lceil \frac{2n-2}{3k} \right\rceil k \equiv 0 \mod 2,$$

then there is an optimal solution to WIP$(n, k, 2)$ *with* $\lceil \frac{2n-2}{3k} \rceil$ *trips. Otherwise we need at most one more trip.*

Proof: Let m be such that

$$3mk/2 + 1 < n \leq 3(m+1)k/2 + 1.$$

We have $m \geq k$, since $n > 3k^2/2 + 1$. Let us suppose that $(m+1)k \equiv 0 \mod 2$, and consider a vertex set $V_{m+1} = \{v_1, v_2, \ldots, v_{m+1}\}$. Vertex v_i identifies with the ith trip. By observation above, there exists a k regular graph, say $\mathcal{G}$, on V_{m+1}. There are $l = (m+1)k/2$ edges in $\mathcal{G}$.

Case 1: $n \equiv 0 \mod 3$. Drop $r = l - n/3 < k$ edges incident on v_{m+1}. So total number of edges in $\mathcal{G}$ is $n/3$. Label each edge in $\mathcal{G}$ arbitrarily with a unique b_j where $1 \leq j \leq n/3$. Now consider vertex labelled v_i. It has $l_i \leq k$ edges incident on it, and each edge has a distinct label. Group these labels arbitrarily into l_i groups of size 1 each, and name these groups $g_i^1, g_i^2, \ldots, g_i^{l_i}$ arbitrarily. Do this for every $1 \leq i \leq m+1$. In all, there are $2n/3$ groups of size 1.

Note that there are still $2n/3$ unused b_j's. Start putting them one by one in $g_1^1, g_1^2, \ldots, g_2^1, g_2^2, \ldots$ until all b_j's are finished. This completes the construction. There are $2n/3$ vertices of degree 1, and rest $n/3$ vertices of degree 2. Moreover there are $2n/3$ hyperedges of size 2 each.

Case 2: $n \equiv 1 \mod 3$. Leave b_n out, and repeat the construction given above for $n-1 \equiv 0 \mod 3$. Now there are 1 vertex of degree 0, $2(n-1)/3$ vertices of degree 1, and rest $(n-1)/3$ vertices of degree 2. Moreover there are $2(n-1)/3$ hyperedges of size 2 each.

Case 3: $n \equiv 2 \mod 3$. Leave b_n out, and repeat the construction given above for $n-1 \equiv 1 \mod 3$. Then the last trip must have $< k$ edges. So pick any vertex, say b_j, that appears in some two earlier trips, but not in the last trip, and add hyperedge (b_j, b_n) to the last trip. Now there are 1 vertex of degree 0 and degree 3 each, $2(n-2)/3+1$ vertices of degree 1, and rest $(n-2)/3-1$ vertices of degree 2. Moreover there are $2(n-2)/3+1$ hyperedges of size 2 each.

The proof of correctness of this construction and the related procedure is left out in this version; we consider the similar problem with larger x in the next section. The construction there is a generalization of the construction given above. We gave $x=2$ construction separately because it is much simpler, and provides a starting point for the general construction. Also, this construction is optimal for all $n > 3k^2/2+1$ if $k\left\lceil \frac{2n-2}{3k} \right\rceil \equiv 0 \mod 2$. Otherwise we need to consider the graph $\mathcal{G}$ on $m+2 = \left\lceil \frac{2n-2}{3k} \right\rceil + 1$ ($\equiv 0 \mod 2$) number of vertices and do similar construction. This gives the overhead of at most one trip more than the optimal. □

The result above considers the case when $k = O(\sqrt{n})$. What if k were larger? Also, the optimal 2-trip procedure needs $k = \left\lfloor \frac{n-1}{2} \right\rfloor$. What if we restrict k to something smaller? Of course, for a fixed n, the number of required trips will increase as k decreases. What is the threshold value of k at which the number of trips jumps from 2 to 3, or from 3 to 4, and so on? Formally, let $K(t,n)$ denote *the minimum k for which it is possible to solve the WIP problem in t trips with* $x=2$. In what follows, we avoid floors and ceilings to keep the formulas simple. It is a simple but tedious matter to modify the expressions to be exact. Interestingly, while the thresholds for going from t to $t+1$ are of the same general type for $t \geq 3$, it is of a different type for $t=2$.

THEOREM 4 *Consider any n > 2. Then,*

$$K(2,n) = (1 - \frac{\Theta(1)}{\sqrt{n}})\frac{n}{2}.$$

For any constant t > 2,

$$K(t,n) = (1 - \frac{\Theta(1)}{\lg n})\frac{n}{t}.$$

Proof: For a given n, we have to find the minimum k such that a connection graph with t labels can be constructed with trivial automorphism group.

Suppose $t=2$. It follows easily from our characterization that the connection graph in this case is a union of paths of even (possibly 0) length with edges alternately labelled 1 and 2. In order to achieve the minimum k, we would like as many paths as possible of *different* lengths. Clearly, we can have at most $O(\sqrt{n})$ different lengths, and we can get close to this by taking the paths of lengths $0, 2, 4, \ldots$. Towards the end of this process we may have the problem that we do not have enough vertices to include a full path. In this case we just join a path on the remaining vertices to the maximum length path already included. If this path is of odd length, then we also connect the included path of length 0 in this path to get a path of even length. It is easy to see that the labelling of the edges can be arranged so that about the same number of labels of each type are used. This proves that $K(2,n) = (1 - \frac{\Theta(1)}{\sqrt{n}})n/2$.

Now we consider the case $t > 2$. The situation here is similar to the $t=2$ case but now the connection graph is a more general graph than the union of paths. We focus on the case $t=3$; larger t is handled similarly. Fix $n > 2$ and consider a connection graph with minimum k. Let us classify its connected components as *tree* and *non-tree*, depending on whether the component is a tree. If the number of the tree components is c, then the number of edges used in the graph is at least $n - c$. We will upper and lower bound the number of tree components that a connection graph with t labels for the edges has, and from that we derive lower and upper bounds on k.

First we prove the upper bound on k. We will restrict the trees to be labelled paths of length $3 \lg n$ (number of edges). Let us number the edges of the paths by $0, 1, \ldots, 3 \lg n - 1$. Consider the following type of labelling of the edges by 1, 2 and 3. For i any nonnegative integer, edges numbered $3i$ get label 1, edges numbered $3i + 1$ and $3i + 2$ get labels $2, 3$ or $3, 2$.

Thus in any such labelling the first vertex of a path is on an edge labelled 1, and the last vertex is on an edge labelled 2 or 3. The number of such labelled paths is $2^{\lg n} = n$. The test graphs of these labelled paths are nonisomorphic and have trivial automorphism groups, and all paths use the same number of labels of each type. Hence, taking the (vertex) disjoint union of $n/(3 \lg n)$ different labelled paths (one of the paths may be of length less than $3 \lg n$) we get a connection graph which solves the problem. The total number of edges in this graph is $n - n/(3 \lg n)$. Since the number of edges with each label is at most one more than one third of this number (this is because of the possibility of the path which is not of length $3 \lg n$). So we can take $k \leq n/3(1 - 1/(3 \lg n)) + 1 \leq n/3(1 - 1/(4 \lg n))$ (for large enough n).

Now for the lower bound on k. If a test graph is nonautomorphic, then so is its underlying connection graph. Here we will only require that the tree components in the connection graph be nonautomorphic and pairwise nonisomorphic. This is only a necessary condition for the test graph to be nonautomorphic, and is in general not sufficient. As remarked above, we prove the lower bound on k by showing an upper bound on the number of tree components a nonautomorphic connection graph can have. The lower bound obtained on k using this condition can be no larger than the actual lower bound.

Recall that since we are working with the case $t = 3$, the edges of our connection graph are labelled by labels 1, 2 and 3 with no edges with the same label being adjacent. To construct a connection graph with the largest possible number of tree components as above we start with edge-labelled trees of size 1 and keep including more and more edge-labelled trees of smallest possible size which have trivial automorphism group and are not isomorphic to a previously included tree. In the end, it may not be possible to add any more trees because not enough vertices are left. It is clear that the graph constructed this way has the largest possible number of tree components and is nonautomorphic. Actually, we have to do a little more in order to keep k minimum possible. We do this by keeping the number of edges with labels $1, 2, 3$ nearly the same, and so about $(n - c)/3$ (c is the number of tree components).

For edge-labelled nonautomorphic trees of size r, we can divide them into disjoint groups of size $3!$, where the labelling of a tree in a group is the same except the labels have been permuted in all possible ways. Since our trees are nonautomorphic, the trees in a group cannot be isomorphic, and thus all the groups are of size 6. Now, in the above procedure when including trees, we include tree in a group together. This keeps the number of edges of each type balanced, except perhaps at the last stage when we may not be able to include a complete group. But the effect of this will be negligible.

The number of distinct unlabelled trees (neither the edges nor the vertices are labelled) with r vertices is asymptotically equal to $U_r = c_1 c_2^{-r} r^{-5/2}(1 + o(1))$, where $c_2 = 0.3383219\ldots$, and $c_1 = 0.53494\ldots$ are constants ([7], see also [6]). Using it, we estimate the number of edge-labelled trees. Our trees have maximum degree upper bounded by 3, but the above more generous bound is good enough for our purposes. It follows that the number of distinct nonautomorphic edge-labelled trees of size r is at most $U_r 3^{r-1}$, as we can label the edges of a tree on r vertices in 3^{r-1} ways. Let $f(r)$ denote the sum of the sizes of nonautomorphic edge-labelled trees of size $\leq r$ with unlabelled vertices and edges labelled by 1 or 2. And let $g(r)$ denote the number of such trees. Then it follows from the above that $g(r) < f(r) < 10^r$ for $r > 1$. So if r is the maximum size of a tree included in the above construction, then we should have either $f(r) = n$ or $f(r-1) < n$ and $f(r) > n$ and so $r > \lg n / \lg 10$. The construction in the upper bound proof above shows that $r \leq 4 \lg n$ and $g(r) > 2^{r/4}$. Since, $g((\lg n)/2 \lg 10) < f((\lg n)/2 \lg 10) < 10^{(\lg n)/2 \lg 10} = \sqrt{n}$, the remaining $n - \sqrt{n}$ vertices are covered by trees of size $> (\lg n)/2 \lg 10$, and thus the number of such trees is at most $(n - \sqrt{n})/(\lg n / 2 \lg 10)$, and at least $(n - \sqrt{n})/(4 \lg n)$. Thus the total number of trees used by the above procedure is $\Theta(n / \lg n)$.

It remains to specify what do we do in the last stage of the procedure when some vertices remain but not enough to add an admissible tree. In such a case we just add a minimal (in the number of edges) edge-labelled graph (not necessarily a tree) on this last set of vertices, which is clearly of size $O(\lg n)$. This is the only place where we do not have control over the relative number of edges with label 1 and 2, but since the number of edges needed here is at most

$O(\lg n)$ (we can always construct a nonautomorphic graph with linear number of edges) this is insignificant compared to the total number of edges which is linear in n.

Thus the number of edges in the graph constructed by the above procedure is $n - \Theta(n/\lg n)$, and each label has $n/3 - \Theta(n/\lg n)$ edges. □

4. *General Case*

In this section, we derive tight lower and upper bounds for WIP(n, k, x) with wide ranges of parameters n, k and x.

Lower Bound. We first show the lower bound.

THEOREM 5 *(Lower Bound) Fix $n, k > 0$, $x > 1$, and suppose that a procedure $\mathcal{P}$ solves WIP correctly. $\mathcal{P}$ needs at least $\left\lceil \frac{2n-2}{k(x+1)} \right\rceil$ number of trips. This lower bound is tight.*

Proof: Since at most k hyperedges are allowed per trip, following observations are immediate. First, $t^{\mathcal{P}} \geq E^{\mathcal{P}}/k$. Second,

$$E^{\mathcal{P}} \geq \frac{2n-2}{x+1}.$$

This can be proved quite simply. Every hyperedge contains at most x vertices, and there are $d_i^{\mathcal{P}}$ vertices of degree i. Therefore,

$$xE^{\mathcal{P}} \geq \sum_{i=1}^{t^{\mathcal{P}}} i d_i^{\mathcal{P}}.$$

By an earlier observation, $E^{\mathcal{P}} \geq d_1^{\mathcal{P}}$. Adding these two inequalities, $(x+1)E^{\mathcal{P}} \geq 2\sum_{i=1}^{t^{\mathcal{P}}} d_i^{\mathcal{P}}$. By earlier observation, $d_0^{\mathcal{P}} \leq 1$. Therefore, $\sum_{i=1}^{t^{\mathcal{P}}} d_i^{\mathcal{P}} \geq n-1$. The result now follows.

As a consequence of above two propositions, the lower bound follows.

We now show that this bound is tight. Let $T : \mathbb{N}^2 \to \mathbb{N}$ and $N : \mathbb{N}^2 \to \mathbb{N}$ be defined as follows: $\forall a, b \in \mathbb{N}$, $T(a,b) = a(b-1)+1$, and $N(a,b) = T(a,b) \cdot a(b+1)/2 + 1$.

Let us fix $k > 0$ and $x > 1$. We now demonstrate a solution for $n = N(k,x)$ wires that consists of $t = T(k,x) = \frac{2n-2}{k(x+1)}$ trips. In fact, we show how to construct a connection hypergraph on n vertices that has total kt hyperedges of size x each and only trivial automorphism group.

Construction: Start with a vertex set $V_t = \{v_1, v_2, \ldots, v_t\}$. Here vertex v_i identifies with the ith trip. Now consider complete graph K_t on V_t. Note that there are $l = \binom{t}{2} = (k(x-1)+1)k(x-1)/2$ edges in the graph. Label each edge arbitrarily with a unique b_j where $1 \leq j \leq l$.

Now consider vertex labelled v_i. It has $k(x-1)$ edges incident on it, and each edge has a distinct label. Group these labels arbitrarily into k groups of size $x-1$ each, and name these groups as $g_i^1, g_i^2, \ldots, g_i^k$ arbitrarily. Then let jth hyperedge of ith trip, namely $e_i^j = \{b_{l+(i-1)k+j}\} \cup g_i^j$ for $1 \leq j \leq k$. Do this for every $1 \leq i \leq t$. This completes the construction. There is 1 vertex of degree 0, kt vertices of degree 1, and the remaining $l = n - kt - 1$ vertices of degree 2. Moreover there are kt hyperedges of size x each.

The proof of correctness is as follows. Any automorphism of this connection hypergraph must map b_n to itself since it is the only vertex with degree 0. Same is true for any degree 2 vertex since every pair of trips has precisely one distinct vertex in common. Now it follows that every degree 1 vertex must also be mapped to itself since its $x-1$ (> 0) neighbors are all degree 2 vertices, and they are being forceably mapped to themselves. Therefore only trivial identity automorphism can exist for the connection hypergraph we have constructed above.[6] □

Upper Bound. We provide a nearly tight upper bound in general.

[6] The proof of correctness in effect gives a simple and practical algorithm for the electrician to identify matching.

LEMMA 6 *Fix $k > 0$ and $x > 2$. Let m be a natural number satisfying $m \geq T(k,x) - 1$ and $(m+1)k(x-1) \equiv 0 \mod 2$. Then there is a $m+1$ trip solution for WIP instance having n wires, where*

$$mk(x+1)/2 + 1 < n \leq (m+1)k(x+1)/2 + 1.$$

Proof: As before, we show how to construct the underlying connection hypergraph with trivial automorphism group.

Consider a vertex set $V_{m+1} = \{v_1, v_2, \ldots, v_{m+1}\}$. As before vertex v_i identifies with the ith trip. As observed earlier, there exists a $k(x-1)$ regular graph, say $\mathcal{G}$, on V_{m+1}. There are $l = (m+1)k(x-1)/2$ edges in $\mathcal{G}$. Label each edge in $\mathcal{G}$ arbitrarily with a unique b_j where $1 \leq j \leq l$. Consider vertex labelled v_i. It has $k(x-1)$ edges incident on it, and each edge has a distinct label. Group these labels arbitrarily into k groups of size $x-1$ each, and name these groups $g_i^1, g_i^2, \ldots, g_i^k$ arbitrarily. Do this for every $1 \leq i \leq m+1$. Observe that there are still $n-l$ unused b_j's. Put them one by one in $g_1^1, g_1^2, \ldots, g_2^1, g_2^2, \ldots$ until all but one b_j's are finished. This completes the construction.

By construction, there is 1 vertex of degree 0, $n-l-1 \leq (m+1)k$ vertices of degree 1, and remaining $l = (m+1)k(x-1)/2$ vertices of degree 2. Moreover there are $n-l-1$ hyperedges of size x each, and $(m+1)k(x+1)/2+1-n$ hyperedges of size $x-1$ each. The proof of correctness of this construction and the related procedure is identical to the one given for the tight example in the lower bound.

□

We can now conclude:

THEOREM 7 *Suppose we are given integers $n, k > 0$, and $x > 2$, that satisfy $n > k^2(x^2-1)/2 + 1$. If*

$$\left\lceil \frac{2n-2}{k(x+1)} \right\rceil k(x-1) \equiv 0 \mod 2,$$

the construction above optimally solves WIP(n,k,x) using $\lceil \frac{2n-2}{k(x+1)} \rceil$. Otherwise it needs at most one more trip.

5. *The Hierarchical Case*

We sketch a $O(\lg^* n)$ solution to WIP when we put the hierarchical constraint on the test sets. First, the upper bound. We show that in two trips we can partition the wires into sets of size $O(\lg n)$, so that each such sets is distinguished from others, and hence we can recurse on each of these sets. This immediately gives the desired $O(\lg^* n)$ solution.

In the first step, we divide the wires into $n/4\lg^2 n$ sets of size $4\lg^2 n$. Next we partition the wires in each of these sets using partitions from the following families. Parts in these partitions come from the pairs: $\{0, 4\lg n\}, \{1, 4\lg n - 1\}, \{2, 4\lg n - 2\}, \ldots, \{2\lg n, 2\lg n\}$. Parts in each pair sum to $4\lg n$, and these parts are distinguishable from each other. Partitions consist of $\lg n$ distinct pairs from the set of above pairs. The number of such parts is at least $\binom{2\lg n}{\lg n} > n$. So we can choose a unique partition for each of the $n/4\lg^2 n$ sets, completing the proof. At the high level, this proof relies on the observation that we do not need a lot of groups of different sizes; instead we need the *set* of sizes that a given group is partitioned into to be different from the sets generated from partitioning other groups. Our algorithm constructs a large number of such partitions while keeping the sum of their sizes equal. The sketch above suffices for the reader to construct a full proof.

Now we sketch the lower bound. We show that a $\lg^* n/3$ step solution is not possible. Suppose, for contradiction, that we have such a solution. For the hierarchical case, it is useful to think of a solution in terms of a tree, where the root of the tree represents the set of n wires; nodes at level 1 (root is at level 0) are the sets at step 1, and so on. One of the following two cases has to occur: (1) All children of the root are of size at most $\lg\lg n$; (2) there is a set of size more than $\lg\lg n$.

In the first case we show that it is not possible to distinguish between all the children, which is a contradiction. In this case, children have at most $\lg\lg n$ different cardinalities. So for at least one such cardinality c there will be at least $n/(\lg\lg n)^2$ children with size c. Since the number of rooted trees on m vertiecs is less than m^m, the number of different rooted trees (actually we have some extra constraints, but that only works to our advantage in this argument) that one can have on $c \leq \lg\lg n$ vertices is $(\lg\lg n)^{\lg\lg n} << n/\lg\lg^2 n$ (for large enough n). So it is not possible to distinguish between the children of size c. Contradiction.

Since case (1) cannot happen, second case always happens. So at level i there is a child of size at least $\lg^{(2i)} n$. So the tree has height at least $(\lg^* n)/3$, contrary to the assumption we started with. This sketch is not a complete proof because we did not consider the case when n is a small constant when the above estimates may break down. This is the reason we claim the lower bound of $(\lg^* n)/3$, instead of $(\lg^* n)/2$. These missing details are not very interesting and easy to supply.

THEOREM 8 *WIP(n, n, n) has solution with $\Theta(\log^* n)$ trips in which all groups tested are hierarchical.*

In contrast, the classical Graham-Knowlton version is non-hierarchical and has solution with 2 trips. Thus there is a separation between the hierarchical and non-hierarchical cases.

6. *Concluding Remarks*

Even though the variant of WIP considered in this paper allowed us to use sets having size between 2 and x, we employed sets of only two different sizes, namely x and $x - 1$, in the construction while our general result. The construction there can be easily modified so that all sets are of size precisely x. Thus, our results hold even when at most k sets of size precisely x are allowed per trip.

Yet another variant is that at least $n - c$ wires must remain *unconnected* per trip. Therefore the only restriction is $x \leq c$. By treating this as a $k = \lfloor c/2 \rfloor$ and $x = 2$ case in our original variant, we get a solution requiring $\frac{4n-4}{3c} + O(1)$ number of trips for $n = \Omega(c^2)$. This is not far from truth. In fact, one can show a matching lower bound of $\lceil \frac{4n-4}{3c} \rceil$ by arguing similar to proofs in this paper.

There are a number of problems we have left open. A technical problem is to determine $K(t, n)$ for general x (we only solved the $x = 2$ version here). An interesting variation of this problem is to solve it when some of the tests are faulty. Group testing in presence of errors is a well-studied topic [8], but errors in the WIP can be quite rich: false positives and false negative for each group tested in each trip. This makes the problems quite challenging.

Acknowledgment: We thank Michal Koucký for a conversation which started this work.

References

[1] D. Du and F. Hwang. Combinatorial group testing and its applications. World Scientific Publication Co, 2nd Edition, 2000.

[2] R. L. Graham and K. C. Knowlton, Method of Identifying Conductors in a Cable by Establishing Conductor Connection Groupings at Both Ends of the Cable, U.S. Patent 3,369,177 (13 Feb. 1968).

[3] R. L. Graham, On partitions of a finite set, *J. Combinatorial Theory*, **1**, (1966), 215–223.

[4] Y. L. Yang, Conjugate partitions of a finite set, *Acta Math. Appl. Sinica*, **8**, (1985), no. 2, 245–247.

[5] D. E. Knuth, The Knowlton–Graham Partition Problem, *J. Combinatorial Theory Ser. A*, **73**, (1996), no. 1, 185—189.

[6] A. Odlyzko, Asymptotic enumeration methods, *Handbook of Combinatorics*, vol 2, R. L. Graham, M. Groetschel, and L. Lovasz, eds., Elsevier, 1995, 1063–1229.

[7] R. Otter, The number of trees, *Ann. of Math. (2)*, **49**, (1948), 583–599.

[8] A. Pelc. Searching games with errors - Fifty years of coping with liars, *Theoretical Computer Science* 270 (2002), 71-109.