Problem 5: A Strange Network

Team Germany

Abstract

In the following, we will describe a complete solution to Problem 5.1 and give some partial results for Problem 5.2.

In Problem 5.1, the answer is $2\binom{n-1}{k} - \binom{n-2}{k} + 1$: Carl can always restore the original code if and only if it's guaranteed that for any two entries in the original subcode, there will be one subcodes among the chosen set of $T_{\min}$ subcodes which contains both of them — and the number $2\binom{n-1}{k} - \binom{n-2}{k}$ counts the number of k-subcodes of an n-subcode not containing both of two fixed entries.

In Problem 5.2, we only found three smaller results: First, we proved that the value of $T_{\min}(n, k, \alpha)$ doesn't depend on α. Then we showed that for infinitely many cases (even for almost all n if k is fixed), the number $T_{\min}(n, k, \alpha)$ is not defined because even if Clara sends all possible subcodes to Carl, then there are two different input codes which yield the same output. This is shown by a simple counting argument. Finally, we proved that for any integer $n > 1$, we have $T_{\min}(n, n-1, \alpha) = \lfloor n/2 \rfloor + 2$.

Problem 5.1

Answer: We have $T_{\min}(n,k,\alpha) = 2\binom{n-1}{k} - \binom{n-2}{k} + 1$.

Proof. Why is $T_{\min} > 2\binom{n-1}{k} - \binom{n-2}{k}$*?* Consider a network which only sends back $2\binom{n-1}{k} - \binom{n-2}{k}$ subcodes. Then it can happen that it sends back only those subcodes where at most one of the numbers a_1 and a_2 appear (a simple counting argument shows that the number of such subcodes is exactly $2\binom{n-1}{k} - \binom{n-2}{k}$), and so Carl doesn't know in which order those two numbers occur in the original code.
Why is $T_{\min} \leq 2\binom{n-1}{k} - \binom{n-2}{k} + 1$*?* Consider a network sending back $2\binom{n-1}{k} - \binom{n-2}{k} + 1$ subcodes. As this number is surely greater than $\binom{n-1}{k}$, every number has to occur in one of the subcodes, so (as all the numbers are different) Carl knows all the numbers and just has to figure out their order. But for every two numbers, there has to be at least one subcode where both of the numbers occur (reversing the argument from above), so Carl can figure out the order of those two numbers. Thus, Carl knows the order of the numbers in the code (he can first get a_1 by searching for the number which stands in front of all the other numbers, then a_2 by looking for the number in front of all the other numbers but a_1, ...).

□

Problem 5.2

We have found three partial results, which we will state in the form of lemmas.
First, we ask the question if $T_{\min}(n,k,\alpha)$ depends on α. Of course, for any $\alpha < 1$, we have $T_{\min}(n,k,\alpha) = 0$, because then knows that all a_i have to be nonnegative integers smaller than 1, so he can be sure that the code is $(0,0,\ldots,0)$ without even seeing one subcode. On the other hand, for $\alpha \geq 1$, we have $T_{\min}(n,k,\alpha) > 0$. The following lemma states that all values for $\alpha \geq 1$ are the same:

Lemma 1. For any $\alpha \geq 1$, we have $T_{\min}(n,k,\alpha) = T_{\min}(n,k,1)$.

Proof. Fix n, k and $\alpha \geq 1$, and let $T_1 = T_{\min}(n,k,1)$, $T_\alpha = T_{\min}(n,k,\alpha)$. We obviously have $T_1 \leq T_\alpha$, as Carl has less information in the case that he only knows that $0 \leq a_i \leq \alpha$ for all i.
Now, suppose that $T_1 < T_\alpha$. Then there are two n-codes $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$ with every entry between 0 and α which have T_1 pairwise different k-subcodes $x^1,\ldots,x^{T_1}$ and $y^1,\ldots,y^{T_1}$ such that $x^\ell = y^\ell$ for all

$\ell \in \{1, \ldots, T_1\}$ (where 'different' is defined as in the problem statement). (Otherwise, every output of T_1 k-sucodes would uniquely define the input, and Carl could thus figure out the input, for example by testing all finitely many cases.)
Now, as $x \neq y$, there is an index j such that $x_j \neq y_j$. Set $r := x_j$ and define two length n binary strings $\hat{x}$ and $\hat{y}$ as follows: For every $i \in \{1, \ldots, n\}$, let $\hat{x}_i = 0$ if $x_i = r$ and $\hat{x}_i = 1$ otherwise, and similarly let $\hat{y}_i = 0$ if $y_i = r$ and $\hat{y}_i = 1$ otherwise. (In other words, substitute all entries r by 0 and all the other entries by 1.) Then we have $\hat{x}_j = 0 \neq 1 = \hat{y}_j$, so $\hat{x} \neq \hat{y}$.
Now for any $\ell \in \{1, \ldots, T_1\}$, define $\hat{x}^\ell$ as the string that we get deleting the same entries we deleted from x to get x^ℓ, and define $\hat{y}^\ell$ in the same way for $\hat{y}$. Then as $x^\ell = y^\ell$, we get that $\hat{x}^\ell = \hat{y}^\ell$ for any ℓ (because we can also get $\hat{x}^\ell$ and $\hat{y}^\ell$ by first writing down x^ℓ and y^ℓ and then substituting every entry r by 0 and all other entries by 1). By the definition of 'different', the subcodes $\hat{x}^1, \ldots, \hat{x}^{T_1}$ are still pairwise different.
Now, suppose that Carl knows that $\alpha = 1$ and Clara sends him the string $\hat{x}$. Then the network gives him any T_1 k-subcodes. Suppose that the network chooses the subcodes $\hat{x}^1, \ldots, \hat{x}^{T_1}$. Then we get that *if Carl gets the subcodes* $\hat{x}^1, \ldots, \hat{x}^{T_1}$*, then he thinks that the original code was* x. In the same way we get that if he gets the subcodes $\hat{x}^1, \ldots, \hat{x}^{T_1}$ then, as these are the same as $\hat{y}^1, \ldots, \hat{y}^{T_1}$, he thinks that the original code was y. Contradiction!

□

From now on, let us thus assume that $\alpha = 1$ and set $T(n, k) := T_{\min}(n, k, 1)$.

Note that there are cases where $T(n, k)$ is not defined: This is not only the case for the trivial example $k = 1$, but also for $n = 4$ and $k = 2$, where the codes $(1, 0, 0, 1)$ and $(0, 1, 1, 0)$ have exactly the same k-subcodes — $(0, 0)$, $(1, 1)$, two times $(0, 1)$ and two times $(1, 0)$ — so there is no number of k-subcodes we can select to distinguish them. We now prove that this isn't even rare:

Lemma 2. If k is fixed, then for almost all values of n, $T(n, k)$ is not defined.

Proof. Suppose that $T(n, k)$ is defined. Then in particular, the multisets of k-subcodes of any two n-strings are different: If two n-codes have the same multisets of k-subcodes, then Carl can't distinguish them by any number of k-subcodes he gets from the network.
We have 2^n binary strings of length 1. The number of k-strings (i.e., potential k-subcodes of some string) is 2^k. The multiset of k-subcodes of any string is a multiset of $\binom{n}{k}$ strings of length k. To choose such a multiset, we have

$\binom{\binom{n}{k}+2^k-1}{2^k-1}$ possibilities. Thus, we get that

$$2^n \leq \binom{\binom{n}{k}+2^k-1}{2^k-1}.$$

Now, the left hand side is an exponential function in n, whereas the right hand side is a polynomial (of degree $k \cdot (2^k - 1)$). Thus, for sufficiently large n, the above inequality will not hold any more. □

It would be interesting to know which is, for given k, the minimal n such that $T(n, k)$ is not defined any more. We believe that the bound given in the proof of the lemma is much too large. Perhaps the correct bound is $2k$ (as it is true for $k = 1$ and $k = 2$).

Lemma 3. For any $n > 1$, we have $T(n, n-1) = \lfloor n/2 \rfloor + 2$.

Proof. If $k = n - 1$, then we get the k-subcodes by deleting one of the n digits.

Why can't we have $T(n, n-1) \leq \lfloor n/2 \rfloor + 1$*?* It is sufficient to show that there are two n-strings which have $\lfloor n/2 \rfloor + 1$ different $(n-1)$-subcodes forming the same multisets.
We distinguish two cases:
Case 1: n **is even.** Let $n = 2k$. Define x as the string which has zeroes everywhere except for a 1 at the k-th position, and let y be the string with zeroes everywhere but at the $(k+1)$-th position:

$$x = (0, 0, \ldots, 1, 0, \ldots, 0),$$
$$y = (0, 0, \ldots, 0, 1, \ldots, 0).$$

By deleting the 1 in x we get a subcode with $(n-1)$ zeroes, and we get the same by deleting the 1 in y. By deleting one of the last k zeroes in x, we get a subcode of the form $(0, 0, \ldots, 1, 0, \ldots, 0)$ with $k-1$ zeroes in front of and $k-1$ behind the 1. We get the same subcode k times from y by deleting one of the first k zeroes. Thus, x and y have $k + 1 = n/2 + 1 = \lfloor n/2 \rfloor + 1$ different $(n-1)$-subcodes forming the same multisets.
Case 2: n **is odd.** Let $n = 2k + 1$. Let x be the string which has zeroes at the first k digits and ones at the last $k + 1$ digits, and let y be the string having zeroes at the first $k + 1$ digits and ones at the last k digits:

$$x = (0, 0, \ldots, 0, 1, 1, \ldots, 1),$$
$$y = (0, 0, \ldots, 0, 0, 1, \ldots, 1).$$

Then x has $k+1$ subcodes of the form $(0,0,\ldots,0,1,1,\ldots,1)$ with k zeroes and k ones obtained by deleting one of the ones in the last $k+1$ digits, and y has the same number of subcodes of the same form obtained by deleting one of the first $k+1$ digits. Thus, x and y coincide in $k+1 = \lfloor n/2 \rfloor + 1$ different $(n-1)$-subcodes.

Why do we have $T(n, n-1) \leq \lfloor n/2 \rfloor + 2$*?* It suffices to show that for any two different n-strings x and y, if we consider the multisets of $(n-1)$-subcodes obtained by deleting every single digit of x (respectively, y), those multisets will intersect in at most $\lfloor n/2 \rfloor + 1$ elements.
Suppose that x and y are two n-strings such that x has different subcodes $x^1, \ldots, x^T$ and y has different subcodes $y^1, \ldots, y^T$ such that $x^\ell = y^\ell$ for any $\ell \in \{1, \ldots T\}$. (We want to show that $T \leq \lfloor n/2 \rfloor + 1$; as T is an integer, it suffices to show that $T \leq n/2 + 1$.) For every $\ell \in \{1, \ldots T\}$, let a_ℓ, b_ℓ be the indices such that we obtain x^ℓ from x by deleting the a_ℓ-th digit, and we obtain y^ℓ from y by deleting the b_ℓ-th digit.
As $x \neq y$, there is a minimal index i and a maximal index j such that $x_i \neq y_i$ and $x_j \neq y_j$. We have $i \leq j$, but we can of course have equality in this estimate. Let $x_i = r$ and $y_i = \overline{r} := 1 - r$ as well as $x_j = s$ and $y_j = \overline{s} := 1 - s$. Consider some $\ell \in \{1, \ldots T\}$. At least one of a_ℓ and b_ℓ has to be at most i: If $a_\ell, b_\ell > i$, then the i-th digit of x^ℓ equals 0 and the i-th digit of y^ℓ equals 1, so we can't have $x^\ell = y^\ell$. Similarly, we have to have $a_\ell \geq j$ or $b_\ell \geq j$. Thus, there can't be any index ℓ such that $i < a_\ell < j$ because this would imply that $i < j$ and thus b_ℓ couldn't satisfy the two inequalities $b_\ell \leq i$ and $b_\ell \geq j$.
There can be an index ℓ such that $a_\ell = i$ or $a_\ell = j$, but there are at most two such indices (as all the selected subcodes have to be different). If there are no other subcodes, then we have $T \leq 2 \leq \frac{n}{2} + 1$, so we are done. Thus, we can assume that there is an index ℓ such that a_ℓ equals neither i nor j. Without loss of generality, suppose that $a_\ell < i$. Then the first $i-1$ digits of y^ℓ equal the first $i-1$ digits of y (which are the first $i-1$ digits of x), so we get that

$$y_{a_\ell} = x_{a_\ell+1} = y_{a_\ell+1} = x_{a_\ell+2} = y_{a_\ell+2} = \ldots = x_i = r.$$

Thus, all the entries of y (and thus of x) between the a_ℓ-th position and the $(i-1)$-th position have to equal r. In particular, we get $x_{i-1} = y_{i-1} = r$.
By the same argument as above, there has to be an index ℓ such that $b_\ell < i$ or $b_\ell > j$. But the first case is impossible because this would imply, by the same argument as above, that $x_{i-1} = y_{i-1} = \overline{r}$. Thus, we have $b_\ell > j$ and get, again by the same argument, that $x_{j+1} = y_{j+1} = \overline{s}$; applying the

same argument for the last time, we see that we can't have $a_\ell > j$ for any j. Thus, the only possibilities for a_ℓ are $1, 2, 3, \ldots i$ and j; as all a_ℓ are different, we get that $T \leq i + 1$. Similarly, we get that all b_ℓ are either i or one of $j, j + 1, \ldots, n$. Thus, $T \leq n - j + 2$. Adding these two inequalities yields

$$2T \leq n + 2 + (i - j) \leq n + 2 \quad \Rightarrow \quad T \leq \frac{n}{2} + 1.$$

□