Problem 5 :
A Strange Network

Team : France 3

Abstract

We used ordered-couples as presented in next page to solve the first question, by proving an equivalent condition which can always be satisfied with at least T_{min} subcodes.

However, we could not solve the second question. We changed the definition of α to simplify the exercise, and studied different situations for particular values of α.

For the final question, we studied minimum values such that Carl could have restored the elements of the couple, or the ordering. Finally, we also gave an approximation of the probability that Carl could have restored the code with T k-subcodes.

Notations

Let us remind that i and j are the ranks of a_i and a_j in the code A defined as the n-tuple $A = (a_1, \ldots, a_n)$.

Let $i < j$:

- The number a_i is *before* a_j in A, subsequently a_j is *after* a_i. We say that a_i is *close before* a_j if $j = i + 1$, otherwise we say that a_i is *far before* a_j. Likewise, a_j can be close or far after a_i.
- The couple (a_i, a_j) is *ordered* when a_i and a_j both belong to a same k-subcode among those which Carl received. We will divide the ordered couples into two groups :
  - the couple (a_i, a_j) is *close-ordered* if a_i is close before a_j in A;
  - the couple (a_i, a_j) is *far-ordered* if a_i is far before a_j in A.

  Note that an ordered couple is a 2-subcode of A.
- We can generalize the definition of ordered couples to ordered k-tuples. These will be used in Situation 2. However, k-tuples cannot be close/far-ordered; the close/far relationships between elements of k-tuples will be expressed through ordered couples.
- We will denote a k-subcode as the k-tuple $[a_{i_k}] = [a_{i_1}, \ldots, a_{i_n}]$.
- The number of k-combinations of a set which contains n elements is the number of distinct subsets of this set. This number will be denoted C_k^n essentially in text; the following notation is more practical in mathematical expressions : $\left(\begin{array}{c} n \\ k \end{array} \right)$
- For any real number x, $\lceil x \rceil$ is the *ceiling* of x, i.e. it is the smallest integer not less than x.

Situation 1

Find or estimate the minimal positive integer $T_{min} = T_{min}(n, k, \alpha)$ such that Carl can restore a code A of length n from any T_{min} different k-subcodes of A.

1. Consider the situation when Carl also knows that all a_i's are distinct.

Ordered couples give us the order in which the two elements of the couple appear in A. Since ordering is crucial in this problem, with 1-subcodes we cannot deduce anything except the elements of A.

From one k-subcode Carl can create C_2^k ordered couples, among which $k-1$ are close-ordered. These ordered couples can restore the initial k-subcode, therefore we have not lost information during the process of "converting" a k-subcode into ordered-couples.

Moreover, from the close-ordered couples we can deduce the far ordered ones – we will prove these assertions in the proof of Lemma 1.

Lemma 1 *Carl can restore a code if and only if he knows all its possible close-ordered couples.*

We will first show that it is sufficient, then necessary to know these couples. Thanks to this lemma we will finally be able to calculate T_{min}.

Restoring a code from ordered couples is the same as restoring it with 2-subcodes.

Let the code A of length n be defined by $A = (a_1, \ldots, a_n)$.

Proof of sufficiency If Carl receives the $(n-1)$ 2-subcodes that are close-ordered couples of A, $(a_{i_1}, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{n-1}}, a_{i_n})$, then he knows that $i_1 < i_2 < i_3 < \cdots < i_{n-1} < i_n$. Knowing this order, Carl can restore the code :

$$A = (a_{i_1}, a_{i_2}, ..., a_{i_{n-1}}, a_{i_n})$$

Therefore it is sufficient to know the close-ordered couples of a code A to restore it.

Since k-subcodes are also codes of length k, we have shown that while converting a k-subcode into ordered couples of A, there is no loss of information, since we can easily restore the subcode.

Note that Carl did not have to know that the ordered couples were actually close-ordered couples.

Proof of necessity : Carl needs to know each of the close-ordered couples of A to restore it. That also proves that he cannot deduce any close-ordered couple from others.

Suppose Carl couldn't deduce all close-ordered couples of A from the k-subcodes he received, then a couple (a_i, a_{i+1}) is not ordered yet.

Let the code A' be:

$$A' = (a_1, \ldots, a_{i+1}, a_i, \ldots, a_n)$$

That is to say, the numbers a_i and a_{i+1} have just exchanged positions.

The k-subcodes of A and A' that do not contain both a_i and a_{i+1} at the same time, are the same whether they come from A or A'.

Therefore when not knowing a close-ordered couple there are at least two possible codes from which Carl could have received the same k-subcodes, so Carl could not deduce the code.

We have given proof of Lemma 1.

Calculate T_{min}

From Lemma 1 we conclude that from any T_{min} k-subcodes of A we can deduce all close-ordered couples of A, and that there exists at least one set S of $T_{min} - 1$ k-subcodes from which we cannot deduce one close-ordered couple (a_i, a_{i+1}).

If we add one, we are sure that from the k-subcodes $[a_{i_k}]$ that Carl has not received we can deduce the last one, that means the two elements of (a_i, a_{i+1}) both are in that last k-subcode. Since every number necessarily appears at least once among the k-subcodes Carl received so that he could restore A, T_{min} does not depend on α because all a_i's are distinct.

For any integers n and k such that $k \leq n$, for any α such that $\alpha \geq n - 1$ (otherwise A could not be defined), $T_{min}(n, k, \alpha)$ satisfies this equation :

$$\exists i,\ \sharp\{\, [a_{i_k}] \,:\, (a_i, a_{i+1}) \subset [a_{i_k}] \subset A \,\} = \sharp\{\, [a_{i_k}] \,:\, [a_{i_k}] \subset A \,\} - (T_{min} - 1)$$

$$T_{min} = \left(\begin{array}{c} n \\ k \end{array} \right) - \left(\begin{array}{c} n-2 \\ k-2 \end{array} \right) + 1$$

Situation 2

2. Consider the case that the numbers ai's are not necessary distinct.

To simplify the problem, we changed the definition of α a little.

Carl is given an integer α, such that $0 \leq \alpha \leq n-1$, and every integer between 0 to α appears at least once in the code A.

To begin with, for any α, Carl must at least know all close-ordered couples of A, the proof of the necessity in Lemma 1 applies.

For all α such that $0 \leq \alpha < n$:

$$T_{min}(n,k,\alpha) \geq T_{min}(n,k,n-1)$$

with the case $\alpha = n-1$ being Situation 1

Study of particular values

Case $\alpha = n-1$:

That was Situation 1.

We add that $T_{min}(n,k,n-1)$ was also the number of k-subcodes such that we deduce all C_2^n ordered couples of A, since the far-ordered ones came from the restored code; in fact, from the T_{min} subcodes, we could directly deduce all ordered couples.

Case $\alpha = 0$:

Obviously, if Carl receives $\alpha = 0$, then $T_{min} = 0$, because he knows there are only zeros.

Case $\alpha = 1$

Here, there are only 0's and 1's.

$k = n - 1$ The expression $T_{min}(n, k, n-1)$ from Situation 1 is not correct here. Suppose Clara sent 2 copies of the code of length n defined by

$$A = (0, 1, 0, \dots) = [a_i : \text{ if } \exists k \in N, i = 2k, \text{ then } a_i = 0 \,; \text{otherwise } a_i = 1]$$

If Carl receives :

$$(0, 1, \dots) = (a_1, \dots, a_{n-1}) \text{ and } (1, 0, \dots) = (a_2, \dots, a_n),$$

then Clara could also have sent (1,0,...).

We prove easily that

$$T_{min}(n, n-1, 1) = 3$$

And we have shown that Lemma 1 does not apply in Situation 2.

$k = 2$ We try to see if it is possible with all ordered couples to restore A, i.e. we want to know whether two codes can share the same set of ordered couples, or if two different codes can have the same numbers of combinations of elements to make $(0, 0)$, $(1, 1)$, $(1, 0)$, $(0, 1)$; we will call each of these numbers the number of the according couple.

Peculiar situation : p is an integer such that $p > 2$. If $n = 2p$ – i.e. n is an even integer strictly greater than 2 – and there are p 0's and p 1's in the code A of length n, then there exists another code A' such that if Carl has received all 2-subcodes when Clara sent enough copies of A, he would be unable to tell if the code sent by Clara was A or A'.

The process is to replace the 1's of A with 0's, and 0's with 1's, it becomes A_1, then the symmetric of that code would be A'.

The numbers of couples $(0, 0)$, $(1, 1)$, $(1, 0)$ and $(0, 1)$ of A are respectively $n_{(0,0)}, n_{(1,1)}, n_{(1,0)}, n_{(0,1)}$; those of A_1 and A' are denoted accordingly.

Because the numbers of 1's and 0's do not change in the entire process :

$$n_{(0,0)} = n_{(1,1)} = n'_{(0,0)} = n'_{(1,1)}$$

First step : $A \mapsto A_1$

We call the image of A its exchanged tuple. Since the positions of 1's and 0's are exchanged :

$$n_{(0,1)} = n_{(1,0),1} \text{ and } n_{(1,0)} = n_{(0,1),1}$$

Second step : $A_1 \mapsto A'$

The image of A_1 is its symmetric :

$$n_{(0,1)} = n_{(1,0),1} = n'_{(0,1)} \text{ and } n_{(1,0)} = n_{(0,1),1} = n'_{(1,0)}$$

Consequently A and A' have the same sets of ordered couples, and Carl cannot deduce A nor A' from 2-subcodes. The order in which the steps were done doesn't change A'.

T_{min} is not defined for $k = 2$.

$A = A'$ iff the exchanged tuple of the first half of A is the symmetric of the second half. In this case, it seems that Carl could find A.

Case $\alpha = n - 2$:

In this case, there is only one repetition. $T_{min}(n, k, n-1)$ k-subcode gives all ordered couples of A.

Then we will know which number was repeated, because we would find an ordered couple $(a_i, a_{i'})$ such that $a_i = a_{i'}$. Moreover, with most codes, there will be ordered couples (a_i, a_j) such that its symmetric (a_j, a_i) is also an ordered couple. This gives a sufficient clue to restore A.

Proof : Because all other a_i's are distinct, we can deduce their ordering in A, that gives us a $(n-2)$-tuple, denoted B.

We can also deduce that the number of ordered couples (a_i, a_j), whose symmetric is also ordered, is the number of distinct elements of ranks j's such that $i < j < i'$. The ordered couples composed of such a_j's give the sequence of a_j between the repeated numbers, which are already set in B. All there is to do then is to place the two a_i's close before and after this sequence.

If there are no ordered couples whose symmetric is also ordered, then (a_i, a_i) is a close-ordered couple, and we restore a code of length $n-1$ as if there was only one a_i, and add another close after this one, we have then restored A.

$$T_{min}(n, k, n-2) = T_{min}(n, k, n-1)$$

Case $\alpha = n - 3$

Here, there would be 2 numbers repeated twice each (1), or 1 repeated 3 times (2). From $T_{min}(n, k, n-1)$ k-subcodes we deduce we are in either case or the other.

(1) We apply the same method as in case $\alpha = n - 2$, ignoring one of the two repeated numbers. The ordered couples of these repeated numbers will be denoted $(a_{i_1}, a_{i'_1})$ and $(a_{i_2}, a_{i'_2})$, the latter being the couple of ignored

numbers. We deduce that we are facing this case if we see two such couples, otherwise we are in case (2).

We then have a code of length $(n-2)$. Applying the previous method once more, we can restore A by inserting two a_{i_2}'s in their positions, relative to non-repeated numbers.

We can notice the cases when among the couples composed with 4 repeated numbers, there are close-ordered couples : either (a_{i_1}, a_{i_2}) or it's symmetric. If we just apply the method of the previous case twice relatively to non-repeated numbers, then there will be at least one a_{i_1} and one a_{i_2} between two same elements, whose order Carl cannot deduce that way.

Ordering these four repeated elements in a 4-tuple will help us place them in the code of length n; but it is not possible with only ordered couples –as shown in case $\alpha = 1$. Consequently T_{min} can be defined only for $k > 2$.

(2) When one number is repeated three times, we can first do as if there were two and restore a code of length $(n-1)$, with first and last repeated elements on positions i_1 and i_3 – the remaining one on i_2.

$k = 2$ **:** That is our most basic case. We do not know if in an ordered couple (a_{i_1}, a_j), a_{i_1} is the first, second or third one. But we know that the Network never gives the same subcode twice., so with $T_{min}(n, 2, n-1)$ 2-subcodes we know that we have all ordered couples of A. Therefore, we know that :

- if $j < i_1$, then we see (a_j, a_{i_1}) three times;
- if $i_1 < j < i_2$, then we see (a_j, a_{i_1}) twice, (a_{i_1}, a_j) once;
- if $i_2 < j < i_3$, then we see (a_j, a_{i_1}) once, (a_{i_1}, a_j) twice;
- if $i_3 < j$, then we see (a_{i_1}, a_j) three times.

  From that we can deduce the rank of a_{i_2}.

Other values of k : Now we cannot count the ordered couples, since one given ordered couple can appear more times than in case $k = 2$. Then we can deduce positions for all elements except the middle a_{i_2}.

We will use ordered 3-tuples in a similar way to place a_{i_2}.

$k = 3$ **:** To know the position of a_{i_2}, we must make sure that in some 3-subcodes we have a_{i_2}, and not a_{i_1} nor a_{i_3}. Besides, to know that the 3-subcodes contains a_{i_2} is not enough, at least one of these codes must contain (a_{i_2-1}, a_{i_2}), and another code, if not the same, must contain (a_{i_2}, a_{i_2+1}), i.e. Carl will have to know between which elements a_{i_2} is.

We will study the 3-subcode which contains (a_{i_2-1}, a_{i_2}). It could be either $(a_j, a_{i_2-1}, a_{i_2})$ (1), or $(a_{i_2-1}, a_{i_2}, a_j)$ (2).

(1) : $(a_j, a_{i_2-1}, a_{i_2})$**.** In this case, the last element could be either a_{i_2} or a_{i_3}, as values they make no differences, but their ranks are different.

Carl would need this 3-subcode to be repeated to know that at least one of them contains a_{i_2}.

(2) : $(a_{i_2-1}, a_{i_2}, a_j)$ If $j \leq i_3$, we are sure this subcode contains the element a_{i_2}.

If $j > i_3$, we will need this subcode to be received twice, as in case (1).

We didn't find any more results.

Additional research

3. Suggest and study additional directions of research.

In this problem, we can note that if Clara wanted to send a message to Carl , if she doesn't know the length of the k-subcodes the Network is going to replace her code with, a simple solution would have been just to send the n elements in ordered couples, so that the Network could only send 2-subcodes (we are assuming the Network at least gives 2-subcodes, otherwise 1-subcodes won't help Carl).

Moreover, we see that T_{min} was calculated using very extreme situations, such as "in no subcode among the $T_{min}-1$ could we deduce one close-ordered couple".

We will try to see if there is a way to minimize the number of copies sent by Clara so that Carl can restore the code most of the time, assuming the Network randomly chooses k.

Situation 1

We noticed α played no role in this situation, which is why we will write :

$$T_{min} = T_{min}(n, k)$$

First, if Carl does not receive enough k-subcodes so that he knows which elements compose A, there is no way he could restore A.

We will study variations of the following values :

- We denote $E(n, k)$ the minimal number of k-subcodes Carl could have received to restore the elements of A.
- We denote $F(n, k)$ the minimal number of k-subcodes Carl must have received so that he is sure to know all elements of A.
- We denote $R(n, k)$ the minimal number of k-subcodes Carl could have received to restore A, i.e. he knows all close-ordered couples.

Even though these values are easy to calculate, we will give proof of their expressions.

We do the Euclidean division of n by k. There exist two integers q and r such that :

$$n = qk + r,\ 0 \leq r < k$$

With $\lceil \frac{n}{k} \rceil - 1$ k-subcodes, there are at most (q-1)k elements of A, which is less than n. Therefore :

$$E(n,k) = \left\lceil \frac{n}{k} \right\rceil$$

To calculate $F(n,k)$, we apply a similar reasoning as the one for T_{min}. With $F(n,k) - 1$ k-subcodes, there is a set of k-subcodes such that there is at least one element Carl has not received, it is one of the sets containing all k-subcodes which do not have one element of A.

$$F(n,k) = \begin{pmatrix} n-1 \\ k \end{pmatrix} + 1$$

One k-subcode can carry at most $k-1$ close-ordered couples. Therefore, applying the same method as for E using ordered couples :

$$R(n,k) = \left\lceil \frac{n-2}{k-1} \right\rceil$$

We can compare F and T_{min} to see if it would be faster for Clara to send copies of a non ordered set so that she is sure that Carl can restore this set. For any n and k :

$$\frac{T_{min}}{F} = \frac{\begin{pmatrix} n \\ k \end{pmatrix} - \begin{pmatrix} n-2 \\ k-2 \end{pmatrix} + 1}{\begin{pmatrix} n-1 \\ k \end{pmatrix} + 1}$$

$$= \frac{[n! - (n-2)!k(k-1) + k!(n-k)!][k!(n-k-1)!]}{[k!(n-k)!][(n-1)! + k!(n-k-1)!]}$$

$$= \frac{n! - (n-2)!k(k-1) + k!(n-k)!}{(n-k)(n-1)! + k!(n-k)!}$$

$$= \frac{n!\left[1 - \frac{(n-2)!k(k-1)}{n!} + \frac{(n-k)!k!}{n!}\right]}{n!\left[\frac{(n-k)(n-1)!}{n!} + \frac{k!(n-k)!}{n!}\right]}$$

In addition, for any k :

$$\lim_{n\to+\infty} \frac{(n-2)!k(k-1)}{n!} + \frac{(n-k)!k!}{n!} = 0$$
$$\lim_{n\to+\infty} \frac{(n-k)(n-1)!}{n!} + \frac{k!(n-k)!}{n!} = 1$$

Therefore :

$$\lim_{n\to+\infty} \frac{T_{min}}{F} = 1$$

But in this limit, k is constant, that means that we showed T_{min} and F are close when n is very big compared to k.

We can prove that although k increases linearly with n, T_{min} gets nearer and nearer to F – the limits shown above would be the same. Then, if Carl wants to be sure to restore each element of a long code without necessarily their ordering, he will have to receive almost as many k-subcodes as if he wanted to know their ordering.

Now we will estimate the probability $P(n,k,T)$ that Carl could have restored all ordered couples from T k-subcodes, $T \leq T_{min}$; we denote $P = P(n,k,T)$.

Obviously, if $T < R$ then $P(n,k,T) = 0$.

The probability P_0 that a k-subcode does not contain a given close-ordered couple is the probability that both of the numbers which compose the couple are not in the k-subcode at the same time – it would be different with far-ordered couples, since you can deduce them from other couples in situation 1.

There are C_k^n possible k-subcodes, and $(2C_k^{n-1} - C_k^{n-2})$ k-subcodes without both numbers.

$$P_0 = \frac{2C_k^{n-1} - C_k^{n-2}}{C_k^n}$$

We notice that :

$$2\begin{pmatrix} n-1 \\ k \end{pmatrix} - \begin{pmatrix} n-2 \\ k \end{pmatrix} = \begin{pmatrix} n \\ k \end{pmatrix} - \begin{pmatrix} n-1 \\ k-1 \end{pmatrix} + \begin{pmatrix} n-1 \\ k-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ k \end{pmatrix}$$
$$= \begin{pmatrix} n \\ k \end{pmatrix} - \begin{pmatrix} n-2 \\ k-2 \end{pmatrix}$$

Then :

$$2\begin{pmatrix} n-1 \\ k \end{pmatrix} - \begin{pmatrix} n-2 \\ k \end{pmatrix} = T_{min} - 1$$

So :

$$P_0 = \frac{T_{min} - 1}{C_k^n}$$

The exact value of the probability P_1 that a given close-ordered couple is not in a set of T k-subcodes is the number of T-combinations of k-subcodes which do not have the close-ordered couple, divided by the total number of T-combinations of possible k-subcodes.

$$\begin{cases} P_1 = 1 - \dfrac{\begin{pmatrix} T_{min} - 1 \\ T \end{pmatrix}}{\begin{pmatrix} C_k^n \\ T \end{pmatrix}}, \text{ if } T < T_{min}; \\ P(n,k,T) = 1, \text{ otherwise.} \end{cases}$$

The Network cannot give two times the same k-subcode. If we take T different subcodes, there is a higher probability of finding the close-ordered couple than if we take T not necessarily different subcodes. The latter process can be interpreted as a Bernoulli process, hence we can give a lower bound of P_1 which is more practical to study :

$$P_1 \geq 1 - (P_0)^T$$

Because the number of all possible k-subcodes fast becomes very important – the following graph shows that $C_6^10 = 141$ – two chosen subcodes are not likely to be the same.

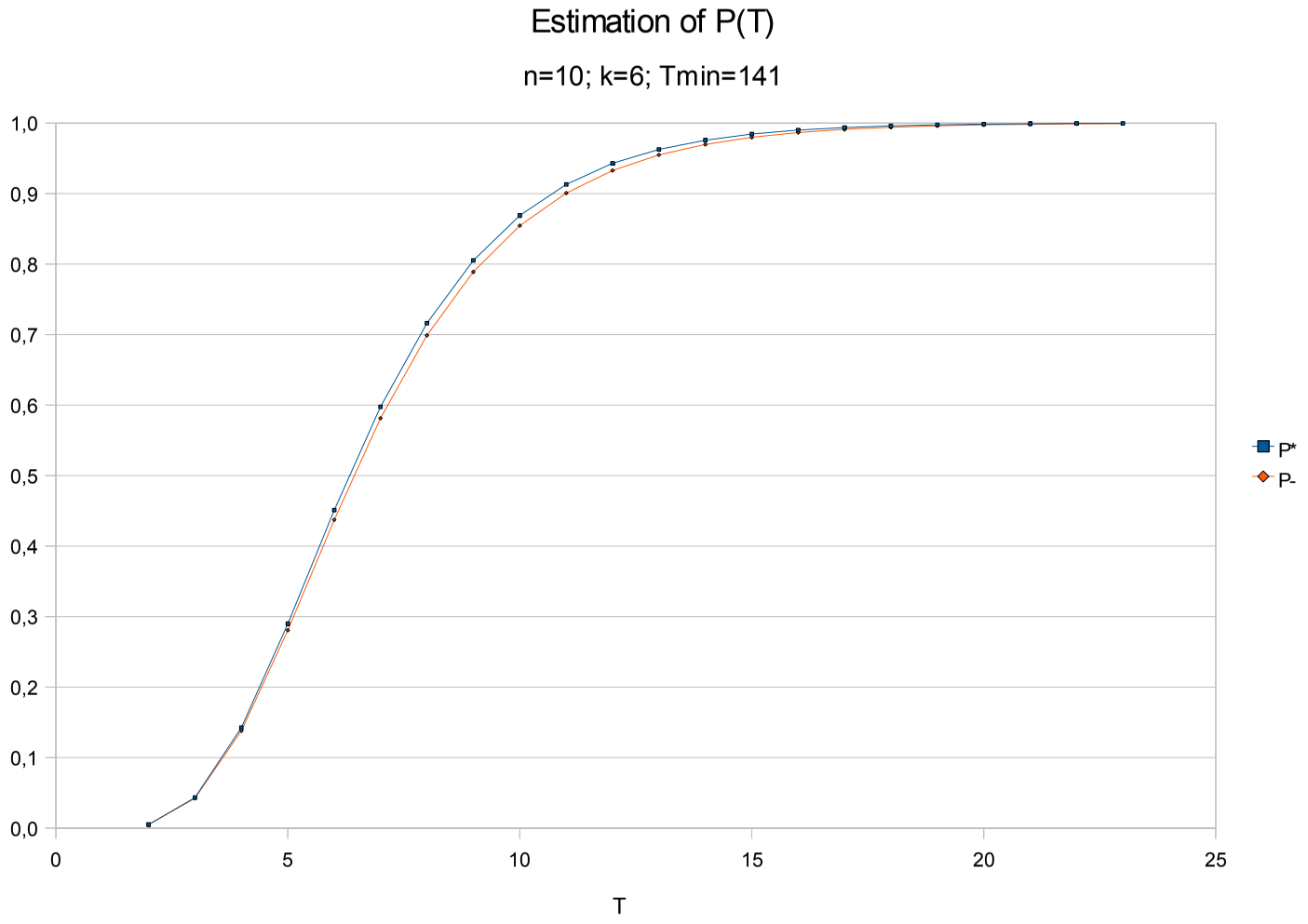
The probability P of finding all $(n-1)$ close-ordered couples can also be simplified as a Bernoulli process. In reality, we think that the fact that our set of k-subcodes contains a close-ordered couple influences the probability that other couples belong to this set. However, the expression of the exact value of P_1 shows that we would be dealing with more combinations of combinations, which is not practical at all, and the lower bound we gave is a fair approximation of P_1.

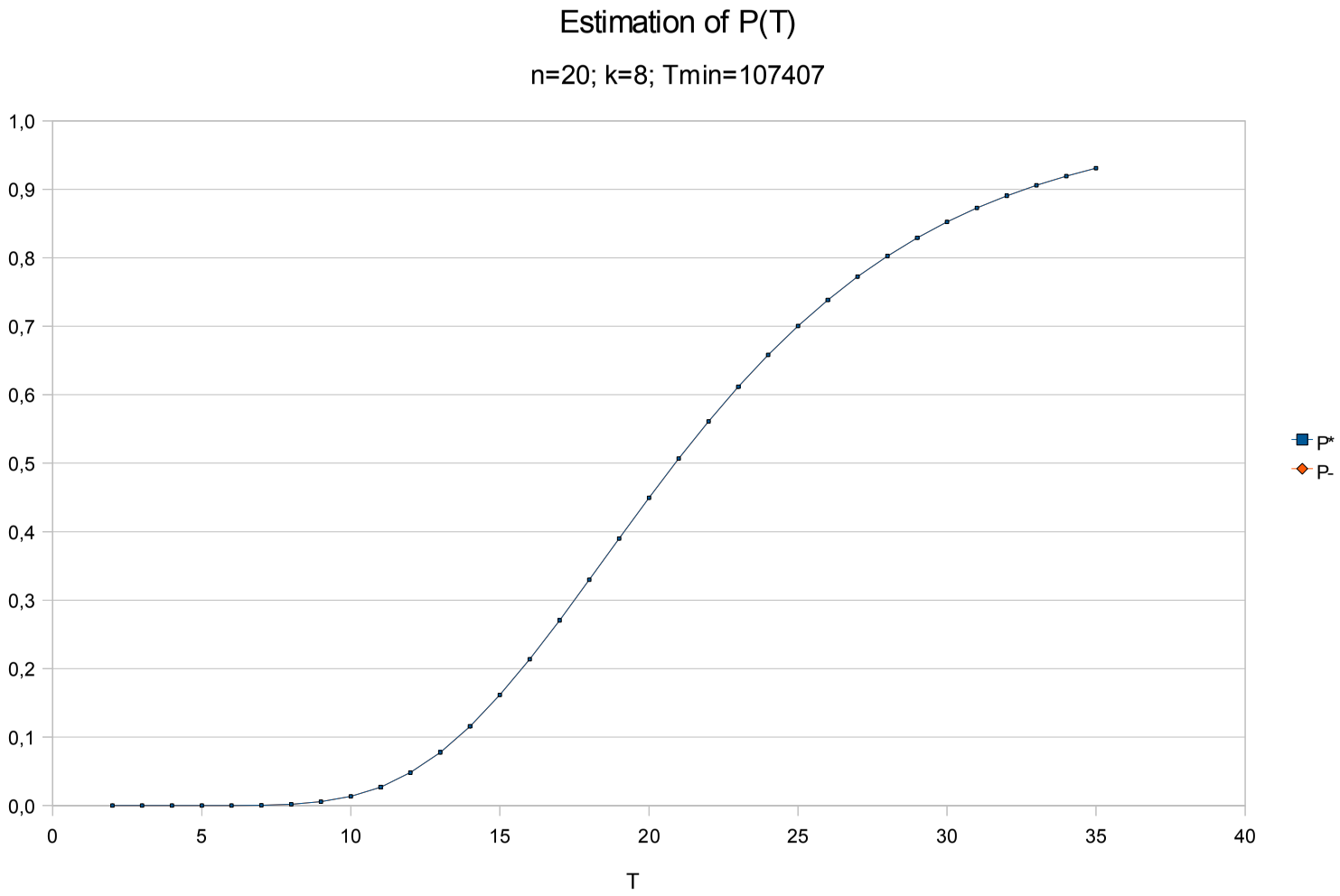
Hence :

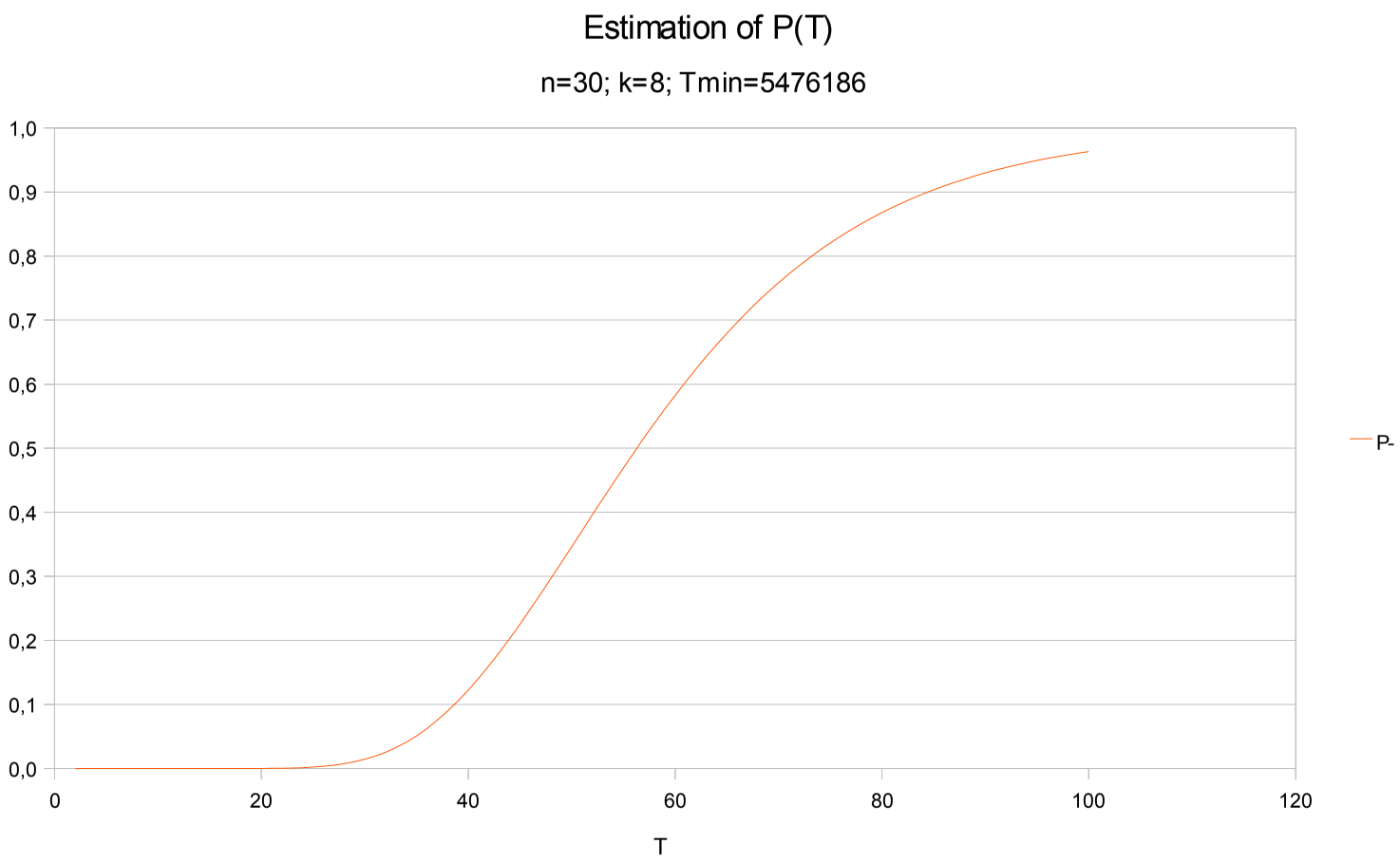
$$\begin{aligned} P(n,k,T) &\approx (P_1)^{n-1} \\ &\approx \left(1 - \frac{\begin{pmatrix} T_{min} - 1 \\ T \end{pmatrix}}{\begin{pmatrix} C_k^n \\ T \end{pmatrix}} \right)^{n-1} && (P*) \\ &\approx (1 - (P_0)^T)^{n-1} && (P-) \end{aligned}$$

The probability that Carl can restore a code of length n with k-subcodes is very high for small values of T compared to T_{min}.

The following graphs show the estimations of P(T) for different values of n and k; T_{min} is given in comparison.


Estimation of P(T)
n=10; k=6; Tmin=141
1,0
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0
5
10
15
20
25
T
P*
P-


Estimation of P(T)
n=20; k=8; Tmin=107407
1,0
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0
5
10
15
20
25
30
35
40
T
P*
P-


Estimation of P(T)
n=30; k=8; Tmin=5476186
1,0
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0
20
40
60
80
100
120
T
P-