Bounds on Maximal Tournament Codes

Brian Gu

Abstract

In this paper, we improve the best-known upper bound on the size of maximal tournament codes, and solve the related problem of edge-covering a complete graph with a minimum number of bipartite graphs of bounded size. Tournament codes are sets of $\{0,1,*\}$ strings closely related to self-synchronizing codes. We improve the current asymptotic upper bound on the size of a length-k tournament code (given by van Lint and van Pul) by a factor in the exponent, and then demonstrate a conditional method for improving the upper bound based on the number of 0 and 1 characters in the optimal tournament code. We also consider a previously-unused equivalence between tournament codes and certain graphs, which relates tournament codes to bipartite coverings of complete graphs. We solve the problem of covering a complete graph with a minimum number of bicliques of bounded size, determining that the minimum number of bicliques of component size x needed to cover a complete graph on n vertices is $\Theta\left(\left(\frac{n}{x}\right)^2+\left(\frac{n}{x}\right)\log x\right)$ (an original result). Finally, we demonstrate the limitations of applying the minimum bounded biclique covering result to the maximal tournament code problem.

1 Introduction

Cryptographers and mathematicians in coding theory often study structures called *codes* because of their uses for data storage, transmission, and encryption. A code is a set of words, whose characters are taken from a given *alphabet*.

Tournament codes, which we study in this paper, are motivated by the properties of *comma-free codes*. Comma-free codes are sets of strings such that the concatenation of any two strings in the code does not have any substrings which are part of the code, besides the two original strings concatenated. Comma-free codes have also been studied by researchers such as Tang [4] to bound maximal tournament codes as early as 1987 [2]. The core property of tournament codes is that there is a *comparability* relation between each pair of strings in the code.

A natural and currently open question which arises from the study of both comma-free codes and tournament codes is the problem of a code's maximal size in terms of the code length. Constructions of Tang [4] and Shor [2] have led to polynomial lower bounds, while other combinatorial arguments have been used to produce super-polynomial upper bounds [5]. In this paper, we consider this problem, improving the upper bound and demonstrating a link between an upper bound and the number of certain characters in the code for future improvements.

We also draw an equivalence between a slightly modified version of the tournament code problem and the problem of edge-covering a complete graph with bipartite graphs. It is known that the minimum number of bipartite graphs needed to edge-cover a complete graph on n vertices is $\lceil log_2 n \rceil$; we show that the minimum number of bipartite graphs with component size at most x is $\Theta\left(\left(\frac{n}{x}\right)^2 + \left(\frac{n}{x}\right)\log x\right)$, and relate this to the modified tournament code problem.

First, in Section 2, we give some mathematical background on the problem and existing work on tournament codes. Next, in Section 3, we use a graph equivalence to tournament codes in order to strengthen the upper bound by a constant factor in the exponent, improving the asymptotic upper bound to nearly the square root of the previous best upper bound. In Section 4, we demonstrate a connection between the number of 0s and 1s in the maximal tournament code and an upper bound on tournament code size, using a correspondence of tournament codes to graph structures. In Section 5 we consider the combinatorially related problem of edge-covering a complete graph with a minimum number of bipartite graphs of bounded size. We prove and characterize an asymptotic formula for the minimum, which we relate to a modified version of the tournament code problem. Finally, in Section 6 we conclude and present possible future directions.

2 Background

A $\{0,1,*\}$ *code* of length k is any set of strings, each of length k, and each composed only of characters from the set $\{0,1,*\}$.

A *tournament code* of length k is a code of length k where every pair of strings in the set is *comparable*. Two strings (s,t) are *comparable* if they satisfy the following properties:

- There are no indices i and j such that either $s_i = 0$, $t_i = 1$, $s_j = 1$, $t_j = 0$ or $s_i = 1$, $t_i = 0$,

0*1100
10*110
*101**

$$\begin{array}{ccc} & 0{*}1100 & \\ \nearrow & & \searrow \\ 10{*}110 & < & {*}101{*}{*} \end{array}$$

Figure 1: A tournament code with length 6 and size 3, with the comparability relationships between them illustrating non-transitivity.

01*
101
1**

$$\begin{array}{ccc} 0 & 1 & * \\ \uparrow & \downarrow & \\ 1 & 0 & 1 \end{array}$$

Figure 2: An example of three strings which are not all pairwise comparable, and which therefore do not form a tournament code. The first and second strings are not comparable because the first is less than the second in the first index, but greater than the second in the second index.

$s_j = 0$, $t_j = 1$. In other words, if s is greater than t at one index, then at no index is s less than t, and vice versa. Note that $*$ is not considered greater or less than either 0 or 1; it is incomparable.

- At some index i, either s is greater than t or t is greater than s.

This definition implies that, for every pair of strings in the tournament code, one is strictly greater than another, and that the greater string has a 1 at some index where the other string has a 0.

Figure 1 is an example of a tournament code of length 6 and size 3. For example, the third string is comparable to (and greater than) the second string because it is greater than the second string at the second character index and at least the second string everywhere else where the two strings are comparable (the fourth character index).

However, the set of strings in Figure 2 is not a tournament code. In Figure 2, the second and third strings are not comparable since there is no index at which one string has a 0 and the other has a 1. Additionally, the first string is greater than the second at position at the second index but less than the second at the first index, violating the comparable property.

Currently, the best lower bound for the maximal size of a k-length tournament code is $k^{3/2}$, based on a construction of Collins, Shor, and Stembridge [2]. The upper bound, given by Tang [4] and by van Lint [5], is asymptotically $k^{\log k}$; more specifically, Levenshtein [3] has shown that for $k \geq 8$, a bound is $k^{\log_4 k}$.

To prove the upper bound, van Lint and van Pul (to whom van Lint attributes the unpublished proof) consider a tournament code as a matrix, where each string represents a row and each character represents an entry. The rows and the columns of the tournament code matrix (of length k and height T) can be permuted without violating the tournament property, as long as the original matrix was a tournament code.

By permuting rows and columns into what van Lint calls *standard form* (Figure 3), van Lint and van Pul are able to generate a recurrence implying the upper bound. To achieve standard form,

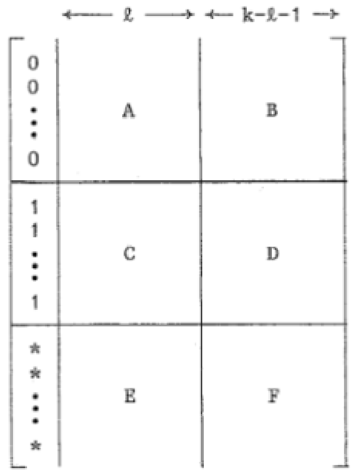

Figure 3: Standard form, as shown in van Lint's [5] diagram. Note that every column in block A has a 1, but that there cannot be any 1s in block B and that there cannot be any 0s in block C.

the rows are first permuted such that every codeword beginning with a 0 is at the top of the matrix, followed by the codewords beginning with a 1. Next, the columns are permuted such that every column in block A has a 1; consequently, no column in block C may have a 0 (or else some string starting with a 0 and some string starting with a 1 would be incomparable), and no column in block B contains a 1. Standard form is referred to again in our improvement on the upper bound.

We often find it helpful to consider a graphical interpretation of the problem. Consider the graph where each string in a given tournament code is considered to be a vertex. If a string s has a 1 at an index where string t has a 0, we direct the edge from s to t on the graph. Because all pairs of strings are comparable, directing edges in this way over all pairs of strings forms a complete tournament on these vertices. This gives rise to the name *tournament codes*, as a round robin tournament is a tournament where between every pair of players there is a win/lose relationship.

Thus, the maximal tournament code problem is equivalent to a graph covering problem. First, consider a column of a k-length tournament code. On the corresponding graph, every string with a 1 in this column has a directed edge towards every string with a 0 in this column, forming a directed bipartite subgraph (Figure 4). Scanning over all columns is then equivalent to covering a complete tournament with directed bipartite subgraphs such that every edge is covered by at least one subgraph and such that no two subgraphs direct an edge in opposite directions. Thus, the problem can be stated equivalently as finding the minimum number of directed bipartite subgraphs needed to cover a complete tournament such that every edge is covered and so that no subgraphs direct an edge in opposite directions.

In considering the graph interpretation of the problem, we use the term *edge-cover* to refer to a covering of a graph by subgraphs such that every edge of the original graph is in one or more of the covering subgraphs. In an edge-covering, overlaps between subgraphs are allowed.

Throughout the paper, $\log x$ by convention refers to the logarithm taken base 2.

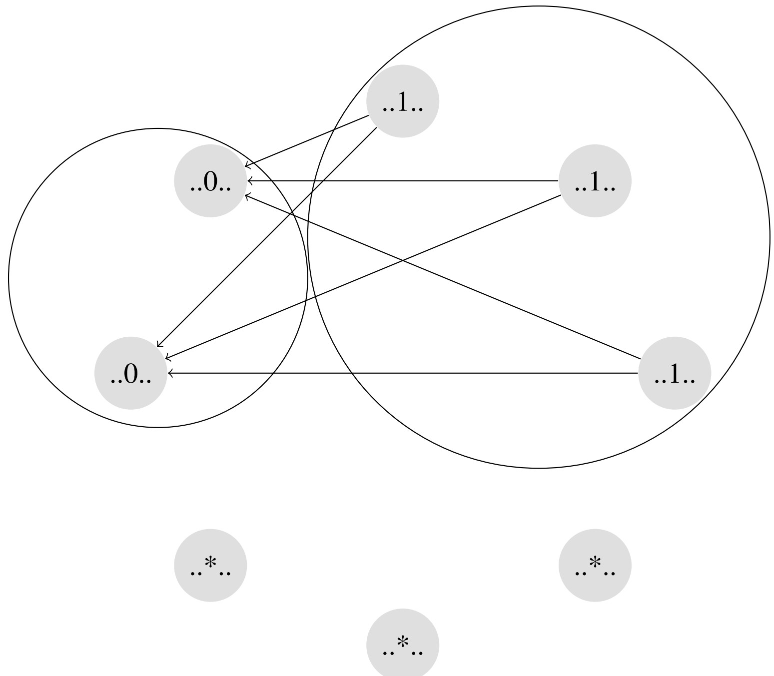

Figure 4: Imposing a bipartite graph on a tournament. Note that, for any given digit, an arrow is drawn from all strings with a 1 in that digit to all strings with a 0 in that digit.

3 Unconditional Improvement on the Upper Bound

We improve the constant factor of the exponent in the Levenshtein's [3] upper bound of $k^{\frac{1}{2}\log k}$. We show in this section that an upper bound is asymptotically $k^{(\frac{1}{4}+\varepsilon)\log k}$, for $\varepsilon > 0$.

First, let $S(k) = k^{1+(\frac{1}{4}+\varepsilon)\log k}$. We make note of two properties of the function S, in Lemma 3.1 and Lemma 3.2. The proofs (which are computational), are omitted due to space constraints.

Lemma 3.1. *If $k > 2^{\frac{1+4\varepsilon}{8\varepsilon}}$, then*

$$S\left(\frac{k}{2}\right)\sqrt{4k} \leq S(k).$$

Proof. The proof of this lemma is computational and given in the appendix. □

Lemma 3.2. *Let $S(k) = k^{1+(\frac{1}{4}+\varepsilon)\log k}$. For all integers a, b, $k_1 \geq 1$, and $k_2 \geq 1$, we have*

$$\min\{aS(k_1), bS(k_2)\} \leq \sqrt{ab}\, S\left(\frac{k_1+k_2}{2}\right).$$

Proof. The proof of this lemma is computational and also given in the appendix. □

In combination with these two properties of the function S, we prove a bound on the number of 0s and 1s in at least one column of the tournament code matrix in order to prove our upper bound.

Lemma 3.3. *Consider a tournament code of length k with maximal size, encoded in a matrix where each entry is a character and each row is a string of characters. Let $x_{0,i}$ be the number of* 0*s in* *column i, and $x_{1,i}$ be the number of 1s in column i. If the width of the tournament code is k and the number of rows in the tournament code is $T(k)$ (which will be referred to as T), then there exists some i such that $x_{0,i}x_{1,i} \geq \frac{T^2}{4k}$.*

Proof. Consider the tournament graph representing the tournament code. The ith column of the tournament code corresponds to a bipartite directed graph imposed on the graph with two components of sizes $x_{0,i}$ and $x_{1,i}$, and a total of $x_{0,i}x_{1,i}$ edges. The bipartite graphs must completely edge-cover the tournament, so $\sum_{i=1}^{k} x_{0,i}x_{1,i} \geq \binom{T}{2}$. Thus the expected number of edges covered by a randomly selected bipartite graph is at least $\frac{T(T-1)}{2k}$, so there exists an index i such that $x_{0,i}x_{1,i} \geq \frac{T(T-1)}{2k}$. It is easy to verify that for $k \geq 1$ and $T \geq 2$ this is at least $\frac{T^2}{4k}$; these conditions always hold. $\square$

With these three lemmas, we can show that S is an upper bound on T, to within a constant factor depending on ε.

Theorem 3.4. *For all $\varepsilon > 0$, there exists c_ε such that $T(k) \leq c_\varepsilon k^{(\frac{1}{4}+\varepsilon)\log k}$.*

Proof. First, we show that $T(k) \leq c_\varepsilon k^{1+(\frac{1}{4}+\varepsilon)\log k} = c_\varepsilon S(k)$. Let c_ε be such that $T(k) \leq c_\varepsilon k^{1+(\frac{1}{4}+\varepsilon)\log k}$ holds for all $1 \leq k \leq 2^{\frac{1+4\varepsilon}{8\varepsilon}}$. Now we show by strong induction that the statement holds true for all $k > 2^{\frac{1+4\varepsilon}{8\varepsilon}}$.

Suppose the statement holds for all $k \leq m-1$. For $k = m$, permute the columns of a maximal-size length-k tournament code such that the first column contains x_0 zeroes and x_1 ones where $x_0 x_1 \geq \frac{T^2}{4m}$. The existence of such a column is guaranteed by Lemma 3.3.

Next, permute the rows and columns of the tournament code into *standard form*, as described by van Lint [5] and as illustrated in Figure 3. In standard form, suppose that block A has length m_1 and block D has length m_2. Since the A and D blocks span $m-1$ columns, $m_1 + m_2 = m-1 < m$. Then the number of strings in block A is bounded above by $T(m_1)$ and the number of strings in block D by $T(m_2)$. Block A also has exactly x_0 strings, while block D has x_1 strings.

If $m_1 = 0$, then no strings in the code with first digit 0 have a 1 as a character. The only way for all strings with first digit 0 to then be comparable is for there to only be one string of first digit 0. Then the strings with first digit either 1 or $*$ form a tournament code of size at most $T(m-1)$, so the maximum length of this tournament code is $T(m-1)+1$. However, we already know that $T(k)$ is super-linear, so it is not possible for the maximal tournament code of length m to have $m_1 = 0$. Thus, $m_1 \geq 1$; a similar argument shows that $m_2 \geq 1$ as well.

Let $x_0 = \frac{T}{a}$ and $x_1 = \frac{T}{b}$. Since $x_0 x_1 \geq \frac{T^2}{4m}$, we know that $ab \leq 4m$. Additionally, we know that $T(m) \leq a x_0 \leq aT(m_1)$ and also $T(m) \leq b x_1 \leq bT(m_2)$. Thus $T(m) \leq \min\{aT(m_1), bT(m_2)\}$. By the inductive assumption, we obtain also

$$T(m) \leq \min\{aT(m_1), bT(m_2)\} \leq \min\{ac_\varepsilon S(m_1), bc_\varepsilon S(m_2)\}.$$

But now by Lemma 3.2, since $m_1 \geq 1$ and $m_2 \geq 1$, we know that

$$T(m) \leq \min\{ac_\varepsilon S(m_1), bc_\varepsilon S(m_2)\} \leq c_\varepsilon \sqrt{ab}\, S\left(\frac{m_1+m_2}{2}\right) \leq c_\varepsilon \sqrt{4m}\, S\left(\frac{m}{2}\right).$$

Finally, by Lemma 3.1,

$$T(m) \leq c_\varepsilon S(m),$$

as desired, so the induction is complete. The theorem follows from the fact that $1 + (\frac{1}{4} + \varepsilon)\log k = O((\frac{1}{4} + \varepsilon)\log k)$. $\square$

Thus, we obtain the new asymptotic upper bound of $k^{(\frac{1}{4}+\varepsilon)\log k}$.

4 Conditional Improvement on the Upper Bound

In Section 3, we improved the upper bound based on a bound for the number of 0s and 1s in the matrix. In this section, we demonstrate a connection between the number of 0s and 1s in a maximal tournament code and an upper bound on its size using a very similar method.

Let x_0 and x_1 be the number of 0s and 1s in a column of a maximal tournament code such that the product $x_0 x_1$ is maximal. Let f be a nondecreasing and positive function of k such that $\log f(k) \log k$ is concave down for $k \geq 1$ and such that

$$x_0 x_1 \geq \frac{T(k)^2}{f(k)}$$

for all k. Note that $f(k)$ is closely related to the proportion of 0s and 1s over the whole code.

Finally, let $S(k) = f(k)^{\log k}$. Note that S is also positive and nondecreasing. As in Section 3, we first prove two lemmas about properties of the function S: a recurrence, and a property from convexity.

Lemma 4.1. *For all positive integers k,*

$$\sqrt{f(k)}S\left(\frac{k}{2}\right) \leq S(k).$$

Proof. Starting from the left hand side,

$$\sqrt{f(k)}S\left(\frac{k}{2}\right) = \sqrt{f(k)}f\left(\frac{k}{2}\right)^{\log \frac{k}{2}},$$

by substitution of S for an expression in f. Noting now that f is a nondecreasing function, we find that

$$\begin{aligned} \sqrt{f(k)}f\left(\frac{k}{2}\right)^{\log \frac{k}{2}} &\leq \sqrt{f(k)}f(k)^{\log k - 1} \\ &\leq f(k)^{\log k} = S(k), \end{aligned}$$

as desired. $\square$

Lemma 4.2. *For all positive real numbers a and b, and for all positive real numbers $k_1 \geq 1$ and $k_2 \geq 1$,*

$$\min\{aS(k_1), bS(k_2)\} \leq \sqrt{ab}\, S\left(\frac{k_1+k_2}{2}\right).$$

Proof. By taking the logarithm of both sides, it is sufficient to show that

$$\min\{\log a + \log S(k_1), \log b + \log S(k_2)\} \leq \frac{\log a + \log b}{2} + \log S\left(\frac{k_1+k_2}{2}\right).$$

It is clear that $\frac{\log a + \log b}{2}$ is at least the minimum of $\log a$ and $\log b$. Now because $\log f(k) \log k$ is concave down for $k \geq 1$, and because k_1 and k_2 are both at least 1,

$$\log S\left(\frac{k_1+k_2}{2}\right) = \log\left(\frac{k_1+k_2}{2}\right) \log f\left(\frac{k_1+k_2}{2}\right) \geq \frac{1}{2}(\log k_1 \log f(k_1) + \log k_2 \log f(k_2))$$

$$= \frac{1}{2}(\log S(k_1) + \log S(k_2)).$$

So

$$\min\{\log a + \log S(k_1), \log b + \log S(k_2)\} \leq \frac{\log a + \log b}{2} + \log S\left(\frac{k_1+k_2}{2}\right),$$

as desired. □

With these properties of S, it is now possible to prove a direct link between the number of 0s and 1s in the maximal tournament code, and the upper bound on the size of the maximal tournament code. The following result relates T and f.

Theorem 4.3. *There exists a constant c_f such that, for all k,*

$$T(k) \leq c_f S(k) = c_f f(k)^{\log k}.$$

Proof. We prove the theorem by the method of strong induction. In the base case, choose c_f so that the statement holds for $k \leq 2$ (which is possible since $S(k)$ is positive). Suppose the statement holds for $1 \leq k \leq m-1$. Now we show that it is true for $k = m$.

First, permute a tournament code with size $T(m)$ into standard form. Suppose that $x_0 = \frac{T(m)}{a}$ and $x_1 = \frac{T(m)}{b}$, and that the lengths of blocks A and D in standard form are m_1 and m_2 respectively. As shown in Section 3, both m_1 and m_2 are at least 1. Then the number of codewords with first digit 0 is at most $T(m_1)$, and the number of codewords with first digit 1 is at most $T(m_2)$. Thus,

$$T(m) \leq \min\{aT(m_1), bT(m_2)\}.$$

And by the inductive hypothesis,

$$T(m) \leq \min\{ac_f S(m_1), bc_f S(m_2)\}.$$

But since $m_1 + m_2 < m$ and S is increasing, by Lemma 4.2 we find,

$$T(m) \leq \min\{ac_f S(m_1), bc_f S(m_2)\} \leq \sqrt{ab}\, c_f S\left(\frac{m}{2}\right) = \sqrt{f(k)}\, c_f S\left(\frac{m}{2}\right).$$

So by Lemma 4.1,

$$T(m) \leq c_f S(m) = f(k)^{\log k}$$

and the induction is complete. □

The significance of this theorem is that it implies that a better asymptotic bound on the number of 0s and 1s in a maximal tournament code results in a better asymptotic upper bound on $T(k)$. In particular, the following corollary demonstrates the implications of proving that the number of 0s and 1s must be linear in the size of the maximal tournament code.

Corollary 4.3.1. *If $f(k) = n$ for some constant n, then $T(k)$ is polynomial in k.*

This corollary implies that, if maximal tournament codes have a number of 0s and 1s linear in the total number of characters, then T is polynomial. As Shor's [2] construction contains a linear number of 0s and 1s and seems quite tight, we conjecture that $T(k)$ is in fact polynomial in k.

5 Minimal Bounded Biclique Coverings

Considering the number of 0s and 1s in maximal tournament codes is akin to considering the size of the bipartite graphs in an edge-covering of a tournament. A natural approach then is to analyze the undirected version of the problem, and to consider the minimum number of bipartite graphs needed to edge-cover a complete graph when the size of the bipartite graphs is bounded.

An *undirected biclique* is a bipartite graph where every vertex in the first set is connected to every vertex in the second set. We use the notation $K_{a,b}$ to denote a biclique where the first set has a vertices and the second set has b vertices. The *size* of the biclique $K_{a,b}$ is defined as $a+b$. Call the two *components* of the biclique $K_{a,b}$ the independent sets of the biclique between which every edge exists.

Minimizing the number of $K_{x,x}$ bicliques in an edge-covering of a complete graph is equivalent to the problem of maximal tournament codes with the following restrictions changed:

1. Instead of requiring pairs of strings to be comparable, we now only require them to be *distinguishable*. Two strings are considered *distinguishable* if there exists an index where one string is 1 while the other string is 0; however, there is no longer any notion of one string being greater or lesser than another. Note that this is a necessary (but weaker) condition for comparability in the original tournament code problem.

2. The number of 1s and 0s in each column is bounded by x.

The main result of this section is the following, an asymptotic expression for the minimum number m of $K_{x,x}$ bicliques needed to edge-cover a complete graph K_n:

Theorem 5.1. *If m is the minimum number of $K_{x,x}$ bicliques needed to cover a complete graph K_n, then $m = \Theta\left(\left(\frac{n}{x}\right)^2 + \left(\frac{n}{x}\right)\log x\right)$.*

We prove this asymptotic expression in several pieces. First, we show that both $\left(\frac{n}{x}\right)^2$ and $\left(\frac{n}{x}\right)\log x$ are asymptotic lower bounds on the number of bicliques needed for a complete edge-covering; this allows us to show that half the sum of the two expressions is a lower bound on m.

The first lower bound is easily shown by counting edges.

Lemma 5.2. *If m is the number of bicliques with component sizes x needed to edge-cover a complete graph on n vertices, then*

$$m \geq \frac{n(n-1)}{2x^2}.$$

Proof. Note that the total number of edges in the complete graph on n vertices is $\frac{n(n-1)}{2}$. Each biclique $K_{x,x}$ has a total of x^2 edges. Thus, there must be at least $\frac{n(n-1)}{2x^2}$ bicliques in an edge-covering of K_n using only $K_{x,x}$. □

To compute the second lower bound, we use the following lemma about the sizes of bicliques.

Lemma 5.3. *Consider an edge-covering of a complete graph G on x vertices using only bicliques. The sum of the sizes of the bicliques is at least $x \log x$.*

Proof. Consider a partial covering of the graph G with c bicliques. We show that if the c bicliques fully edge-cover G, then the sum of their sizes is at least $x \log x$, where x is the number of vertices in G.

Create a matrix M with x rows and c columns, where each row represents a vertex and each column represents a biclique. Enumerate the bicliques $B_1, B_2, \ldots, B_c$. For the ith biclique, consider the two components; for each vertex v in the first component, assign a 0 to the entry in the vth row and the ith column, and for each vertex u in the second component, assign a 1 to the entry in the uth row and the ith column. Assign a $*$ to the remaining entries in the column.

Two vertices have an edge between themselves if there exists a column such that one vertex has a 1 in the column while the other has a 0. Call two vertices *distinguishable* if such a column exists, and *indistinguishable* otherwise. This is equivalent to calling two $\{0,1,*\}$ strings *distinguishable* if there exists some index where one string has a 0 and the other has a 1, and *indistinguishable* otherwise.

Now we use a pigeonhole argument to show that if every vertex is distinguishable, then $\sum |B_i| \geq x \log x$.

Consider 2^c holes, each representing a unique bitstrings of length c. For each vertex v, place a copy of vertex v in every hole where its row string–formed by considering the vth row of M as a $\{0,1,*\}$ string–is indistinguishable from the hole's bitstring.

If two vertices are placed in the same hole, then they are indistinguishable and the partial covering is not a complete edge-covering because indistinguishable vertices do not have an edge between them; otherwise, one would have a 0 at an index where another has a 1, and they would not be indistinguishable with the same $\{0,1\}$ bitstring. Thus, by the pigeonhole principle, the total number of vertex copies placed in holes must be at most the number of holes itself, 2^c.

We now count the number of vertex copies. Suppose that vertex v's row has s_v stars, and that the total number of stars in the matrix is s. Then v's bitstring is indistinguishable from 2^{s_v} bitstrings of length c, so a total of 2^{s_v} copies of v are placed in holes. Thus the total number of vertex copies over all vertices is $\sum_{i=1}^{x} 2^{s_i}$. If the partial covering completely covers G, this sum is at most 2^c. However, by the convexity of the function 2^n, we know that

$$x2^{\frac{s}{x}} \leq \sum_{i=1}^{x} 2^{s_i} \leq 2^c.$$

Therefore

$$x2^{\frac{s}{x}} \leq 2^c.$$

$$2^{\log x + \frac{s}{x}} \leq 2^c.$$

$$\log x + \frac{s}{x} \leq c,$$

which can be rewritten as

$$cx - s \geq x \log x.$$

But $cx - s$ is precisely the number of 0s and 1s in M, and is thus the sum of the sizes of all the bicliques in the complete edge-covering. Thus

$$\sum |B_i| \geq x \log x,$$

as desired. □

We also use a result on the minimum number of bicliques needed to cover a complete graph. This result is well-known; the following proof is adapted from [1].

Corollary 5.3.1. *The minimum number of bipartite graphs of any size needed to cover a complete graph G on x vertices is $\lceil \log x \rceil$.*

Proof. By Lemma 5.3, the sum of the sizes of the bicliques in an edge-covering of G using only bicliques is at least $x \log x$. Because every biclique in a graph with only x vertices can have size at most x, the number of bicliques needed is at least $\frac{x \log x}{x} = \log x$. Since an integer number of bicliques must be used, the number of bicliques needed is at least $\lceil \log x \rceil$.

Now we show that we can use exactly $\lceil \log x \rceil$ bicliques to edge-cover G. Assign to each vertex of G a unique bitstring of length $\lceil \log x \rceil$. Now consider $\lceil \log x \rceil$ bicliques, each generated in the following way:

- For the ith biclique, connect every vertex with a 0 in the ith index of the vertex's bitstring to every vertex with a 1 in the ith index.

Since every bitstring is distinct, for every pair of vertices there exists some index where their digits differ. Thus, every pair of vertices has an edge between them and the $\lceil \log x \rceil$ bicliques completely edge-cover the graph G. □

The following related corollary will be used later to prove the upper bound.

Corollary 5.3.2. *There exists a covering of K_{2x} using $\lceil \log 2x \rceil$ $K_{x,x}$ subgraphs.*

Proof. First, consider any minimal covering of a K_x using the procedure in Corollary 5.3.1, and the associated set of bitstrings B. Now consider C, the set of complements of these bitstrings (bitstrings are complements if they have the same length and do not have the same character at any index). Construct a set B' of the same size as B by appending a 0 to the front of each bitstring in B. Construct a set C' by appending a 1 to the front of each bitstring in C. The intersection of B' and C' is clearly empty, so their union has $2x$ bitstrings. Now consider A, the union of B' and C'. Because B and C are complements, for every index except the first, the number of strings with a 0 at that index in A is equal to the number of strings with a 1 at that index. Finally, by the construction of B' and C', there are an equal number of 0s and 1s in the first index. It follows from here that A encodes a covering of a K_{2x} using $\lceil \log x \rceil + 1 = \lceil \log 2x \rceil$ subgraphs $K_{x,x}$, and that A is a minimal covering of a K_{2x}. □

With Lemma 5.3 and Corollary 5.3.1, we now present a proof that the second term in the asymptotic formula is a lower bound.

Lemma 5.4. *If m is the number of bicliques with component size x needed to completely edge-cover a complete graph on n vertices, then*

$$m \geq \frac{n \log 2x}{2x}.$$

Proof. From Lemma 5.3, the sum of sizes of bicliques in a covering of K_n with only bicliques must be at least $n \log n$. If all of the bicliques have component sizes x, then each biclique has total size $2x$, so there must be at least $\frac{n \log n}{2x}$ bicliques, which is at least $\frac{n \log 2x}{2x}$. □

Finally, we compute an upper bound on m via construction, which is asymptotically equivalent to the lower bound.

Lemma 5.5. *There exists a complete edge-covering of a complete graph on n vertices using at most*

$$4\binom{\lceil \frac{n}{2x} \rceil}{2} + \left\lceil \frac{n}{2x} \right\rceil \lceil \log 2x \rceil$$

complete bipartite graphs $K_{x,x}$ of component size x.

Proof. To impose this upper bound on m, we present the following construction. Let G be the complete graph on n vertices we wish to cover.

We partition the vertices of G into $\lceil \frac{n}{2x} \rceil$ disjoint groups of at most $2x$ vertices each. It takes four $K_{x,x}$ to connect every edge going between any two of the groups; there are $\lceil \frac{n}{2x} \rceil$ groups, so $4\binom{\lceil \frac{n}{2x} \rceil}{2}$

bipartite graphs can be used to cover every edge that goes between all pairs of vertices not in the same group. The remaining uncovered edges now only exist within groups of at most $2x$ vertices each. But by corollary 5.3.2, we know that each of these can be edge-covered with $\lceil \log 2x \rceil$ $K_{x,x}$ bipartite graphs each; since there are $\lceil \frac{n}{2x} \rceil$ of these, our construction requires

$$4\binom{\lceil \frac{n}{2x} \rceil}{2} + \left\lceil \frac{n}{2x} \right\rceil \lceil \log 2x \rceil$$

bipartite graphs to completely edge-cover the K_n graph G. □

Remark. *The following is an upper bound without the ceiling function:*

$$m \leq \frac{(n+2x)(n+x)}{2x^2} + \frac{n+2x}{2x} \log 2x.$$

With the upper and lower bounds shown, we can prove the main result of this section.

Proof of Theorem 5.1. Both the lower and the upper bounds on the minimum are $\Theta\left(\left(\frac{n}{x}\right)^2 + \left(\frac{n}{x}\right)\log x\right)$. Thus the minimum itself is also $\Theta\left(\left(\frac{n}{x}\right)^2 + \left(\frac{n}{x}\right)\log x\right)$. □

The nature of our proofs also allows us to establish a limit to the factor by which the upper and lower bounds may differ.

Remark. *We can also easily show that for all $\varepsilon > 0$, the upper and lower bounds are provably within a factor of $6+\varepsilon$ of each other when n is sufficiently large. Additionally, for all $\varepsilon > 0$, the upper and lower bounds are provably within a factor of $2+\varepsilon$ of each other when $\frac{n}{x}$ is sufficiently large.*

Though this solves the bounded biclique covering problem and with it the equivalent modified maximal tournament code problem, this result also makes it clear that considering the undirected model of the problem is not strong enough to result in a better bound on the number of 0s and 1s. If x is the largest number of 0s or 1s in any column of a maximal tournament code of length k, Theorem 5.1 only gives the following inequalities on x. These inequalities are due to the fact that the asymptotic formula noted in the above remark is also a lower bound on the number of bicliques with components of size at most x needed to cover a complete graph. First,

$$x \geq \frac{c_1 T(k)}{\sqrt{k}},$$

based on the inequality

$$k \geq c_2 \left(\frac{T(k)}{x}\right)^2,$$

for some constants c_1 and c_2. Additionally,

$$x \geq \frac{c_3 T(k)(\log T(k) - \log k)}{k},$$

based on the inequality

$$k \geq \frac{c_4 T(k) \log x}{x},$$

for some constants c_3 and c_4.

Both of these inequalities are at least as weak as the bound on the number of 0s and 1s shown in Lemma 3.3, demonstrating the weakness of the undirected model.

6 Conclusion

We unconditionally reduced the upper bound to nearly the square root of the previous upper bound, and exhibited a method to improve the upper bound further based on the number of 0s and 1s in the maximal-size tournament code. If the number of 0s and 1s is is proportional to the total number of characters in the code, then the maximal tournament code size is polynomial in the length of the tournament code. A graphical interpretation of the problem provides a novel perspective which the string representation masks, allowing us to make these improvements.

We also considered the undirected version of the problem and proved an asymptotic expression for the number of bounded-size undirected bicliques needed to cover a complete graph. We showed that this effectively solves a slightly modified version of the maximal tournament code problem, and demonstrated limitations in applying the biclique result to the original tournament code problem.

Future work on this problem might include trying to impose a bound on the number of 0s and 1s in the tournament code that is stronger than the one shown in Lemma 3.3. Another future step may be bounding the number of 0s in the code relative to the number of 1s, and vice versa. A construction for a better lower bound than $k^{3/2}$ could also be searched for, but it seems difficult to improve this bound as Shor's [2] construction seems very tight. Additionally, as our work with the biclique covering of an undirected complete graph shows, the full power of the comparable property in tournament codes will likely be needed to impose a better bound on the number of 1s and 0s in the tournament code.

7 Acknowledgments

Firstly, I would like to thank Mr. Brandon Tran of the Massachusetts Institute of Technology Mathematics Department for mentoring me and guiding my research.

I would like to thank Dr. John Rickert, Dr. Jenny Sendova, Prof. Andrew Charman, Mr. Shashwat Kishore, Ms. Sarah Shader, and Mr. Richard Yip for looking over my paper from initial stages to the final product, and for their feedback, edits, and improvements to the paper. I would also like to thank Prof. Peter Shor, Prof. Ankur Moitra, and Dr. Tanya Khovanova, Mr. James Tao, Mr. Mark Sellke, and Mr. Yang Liu for their helpful suggestions about how to attack the problem, and Prof. David Jerison, Prof. Ankur Moitra, and Dr. Tanya Khovanova of the MIT Mathematics

Department for coordinating the math program at RSI. Special thanks go to Dr. Tanya Khovanova as well for her valuable advice on the research process, on paper-writing, and on presenting.

Additionally, I would like to thank the Center for Excellence in Education, the Research Science Institute, and my sponsors for offering me the opportunity to begin my research at MIT.

References

[1] Evan Chen. Directed edge chromatic numbers over a tournament. Available at `http://artofproblemsolving.com/community/c6h1065569p4628085` (2015/07/27).

[2] Karen L. Collins, Peter W. Shor, and John R. Stembridge. A lower bound for 0,1,* tournament codes. *Discrete Mathematics*, 63:15–19, 1987.

[3] Vladimir I. Levenshtein. Combinatorial problems motivated by comma-free codes. *Journal of Combinatorial Designs*, 12:184–196, 2004.

[4] Betty Tang, Solomon W. Golomb, and Ronald L. Graham. A new result on comma-free codes of even word-length. *Canadian Journal of Mathematics*, 39:513–526, 1987.

[5] J. H. van Lint. $\{0, 1, *\}$ distance problems in combinatorics. In Ian Anderson, editor, *Surveys in Combinatorics 1985*. Cambridge University Press, Cambridge, 1985.

Appendix A Proof of Lemmas

We present the computational proof of Lemma 3.1.

Lemma. *Let* $S(k) = k^{1+(\frac{1}{4}+\varepsilon)\log k}$. *If* $k > 2^{\frac{1+4\varepsilon}{8\varepsilon}}$, *then*

$$S\left(\frac{k}{2}\right)\sqrt{4k} \leq S(k).$$

Proof. The condition implies that

$$\begin{aligned}
2^{\frac{1}{4}+\varepsilon} &\leq k^{2\varepsilon} \\
k^{\frac{1}{2}}2^{\frac{1}{4}+\varepsilon} &\leq k^{2(\frac{1}{4}+\varepsilon)} \\
2\sqrt{k}\,\frac{2^{\frac{1}{4}+\varepsilon}}{k^{\frac{1}{4}+\varepsilon}} &\leq 2k^{\frac{1}{4}+\varepsilon} \\
\sqrt{4k}\,\frac{2^{\frac{1}{4}+\varepsilon}}{2^{(\frac{1}{4}+\varepsilon)\log k}} &\leq 2k^{\frac{1}{4}+\varepsilon} \\
\sqrt{4k}\,\frac{1}{2^{(\frac{1}{4}+\varepsilon)(\log k-1)}} &\leq 2k^{\frac{1}{4}+\varepsilon} \\
\sqrt{4k}\,\frac{1}{2^{(\frac{1}{4}+\varepsilon)(\log \frac{k}{2})}} &\leq 2k^{\frac{1}{4}+\varepsilon} \\
\sqrt{4k}\,\frac{k^{(\frac{1}{4}+\varepsilon)(\log k-1)}}{2^{(\frac{1}{4}+\varepsilon)(\log \frac{k}{2})}} &\leq 2k^{(\frac{1}{4}+\varepsilon)\log k} \\
\sqrt{4k}\,\frac{k^{(\frac{1}{4}+\varepsilon)\log \frac{k}{2}}}{2^{(\frac{1}{4}+\varepsilon)(\log \frac{k}{2})}} &\leq 2k^{(\frac{1}{4}+\varepsilon)\log k} \\
\sqrt{4k}\left(\frac{k}{2}\right)^{(\frac{1}{4}+\varepsilon)\log \frac{k}{2}} &\leq 2k^{(\frac{1}{4}+\varepsilon)\log k} \\
k\sqrt{4k}\left(\frac{k}{2}\right)^{(\frac{1}{4}+\varepsilon)\log \frac{k}{2}} &\leq 2k^{1+(\frac{1}{4}+\varepsilon)\log k} \\
\left(\frac{k}{2}\right)\sqrt{4k}\left(\frac{k}{2}\right)^{(\frac{1}{4}+\varepsilon)\log \frac{k}{2}} &\leq k^{1+(\frac{1}{4}+\varepsilon)\log k} \\
\sqrt{4k}\left(\frac{k}{2}\right)^{1+(\frac{1}{4}+\varepsilon)\log \frac{k}{2}} &\leq k^{1+(\frac{1}{4}+\varepsilon)\log k} \\
S\left(\frac{k}{2}\right)\sqrt{4k} &\leq S(k),
\end{aligned}$$

as desired. □

We also present the computational proof of Lemma 3.2.

Lemma. *Let* $S(k) = k^{1+(\frac{1}{4}+\varepsilon)\log k}$. *For all integers a, b,* $k_1 \geq 1$, *and* $k_2 \geq 1$, *we have*

$$\min\{aS(k_1), bS(k_2)\} \leq \sqrt{ab}\, S\left(\frac{k_1+k_2}{2}\right).$$

Proof. Note that by taking the logarithm of both sides, it suffices to show that

$$\min\{\log a + \log S(k_1), \log b + \log S(k_2)\} \leq \frac{\log a + \log b}{2} + \log S\left(\frac{k_1+k_2}{2}\right).$$

Observe that $\frac{\log a + \log b}{2}$ is at least the minimum of $\log a$ and $\log b$. Furthermore, since $\log S(k) = (1+(\frac{1}{4}+\varepsilon)\log k)\log k$ is a concave down function for $k \geq 1$ and sufficiently small ε, by Jensen's Inequality $\log S(\frac{k_1+k_2}{2})$ is at least the arithmetic mean of $\log S(k_1)$ and $\log S(k_2)$. Thus $\log S(\frac{k_1+k_2}{2})$ is also greater than the minimum of $\log S(k_1)$ and $\log S(k_2)$, so

$$\min\{\log a + \log S(k_1), \log b + \log S(k_2)\} \leq \frac{\log a + \log b}{2} + \log S\left(\frac{k_1+k_2}{2}\right),$$

as desired. □