Hats Required

Perfect and Imperfect strategies for the Hat Problem*

Megan Breit[1]

Didier Deshommes[2]

Andrea Falden[3]

Abstract. We investigate the hat color problem for both perfect and imperfect team sizes by considering ways of constructing sets of losing configurations and corresponding strategies. We also obtain upper bounds for the number of equivalent sets of losing configurations for both perfect and imperfect team sizes.

1. Introduction

What is an easy way for a mathematician to make \$2 million dollars? Let's play the hat game! This game involves a host, a team of n players ($n > 2$), and two distinct hat colors, say red and blue. While blindfolded, each member of the team is assigned a hat, the color of which is determined by an independent coin toss. Once the hats are in place, the blindfolds are removed, at which point each contestant is able to see the colors of her teammates' hats, but not her own. The object of the game is to have at least one player guess her hat color correctly without any incorrect guesses from the team. So, when the signal is given by the host, all players simultaneously, either guess their hat color or pass. At least one person must guess (that is, the entire team cannot pass); and winning requires that there be no incorrect guesses. No communication is allowed during the game, but the contestants are allowed to meet beforehand to discuss their strategy.[1]

The problem dates back to 1998, when Dr. Todd Ebert, a computer science instructor at the University of California at Irvine, developed a mathematical problem in his Ph.D. thesis that later became a hypothetical game show. Soon after, the puzzle now known as

* The authors acknowledge support received from the National Science Foundation and the National Security Agency via the SUMSRI program administered by Miami University.

[1] Augustana College, Sioux Falls, SD; mjbreit@ole.augie.edu.

[2] North Carolina State University, Raleigh, NC; dfdeshom@unity.ncsu.edu.

[3] Salem College, Winston-Salem, NC; faldeam4@salem.edu.

the hat problem became posted on the Internet, leading to a widespread interest among mathematicians [9]. At first glance, one might be inclined to think that the best any team could do is win half the time since a player can receive no useful information about his hat color by looking at those of his teammates. Further one may propose that an optimal strategy is to have one specific player guess the same color each time while the rest of the team passes. This too yields a winning probability of only $\frac{1}{2}$. However, it has been shown by Dr. Elwyn Berlekamp that in fact, cooperation among team members can significantly increase a team's chances of winning [9].

2. Preliminaries

Let $n > 2$ be the number of players on a team and $q \geq 2$ be the number of distinct hat colors. Then every ordered arrangement of hat colors is called a *configuration*. In the case of 3 players, there are $2^3 = 8$ total configurations because each player has one of two hat colors. The set of all configurations is therefore:

$$\{RRR, BBR, RRB, BRB, RBR, RBB, BRR, BBB\}.$$

If we use 0 to represent red and 1 to represent blue, then our set of configurations is just the vector space F_2^3 over the field $F_2 = \{0, 1\}$:

$$\{000, 110, 001, 101, 010, 011, 100, 111\}.$$

Definition 2.1 A *strategy* S, is a set of deterministic instructions for each player to be executed once the game begins.

Using a given strategy S, the vector space F_2^n is partitioned into a set of losing configurations, denoted L_s, and a set of winning configurations, denoted W_s. That is, $F_q^n = W_S \cup L_S$ and clearly $W_S \cap L_S = \emptyset$.

Definition 2.2 Let $n > 2$ be the number of players on a team. Let S be the strategy employed by the team and let w be the total number of correct guesses made by all n players in all the 2^n configurations. We say S is a *perfect strategy* if $|W_S| = w$ and $|L_S| = \frac{w}{n}$.

Example: If any player sees two of the same color hats on his or her teammates' heads, then that player guesses the opposite color when the cue is given; otherwise, he or she passes. Referring to the configurations for three players, in any of the cases where there are two hats of the same color and one hat of the opposite color, exactly one player will guess correctly and the remaining players will pass. With only one player guessing correctly each time, the correct guesses are spread out, maximizing the order of W_s. In addition, we can see that when each player is wearing a red hat, all players will say "blue" when given the cue. Similarly, if all three hats are blue, then three incorrect guesses of red will be made. Thereby achieving the smallest value of $|L_s|$, in that the incorrect guesses are clustered in to as few configurations as possible. Of the eight configurations, six will result in wins, while only two will result in losses using this perfect strategy. Therefore, the probability of winning is $\frac{6}{8} = \frac{3}{4}$, which is an improvement from $\frac{1}{2}$.

As we saw in [1], perfect strategies correspond directly with perfect codes. In particular, the values of n for which perfect strategies exist are the same values for which Hamming codes exist.

Theorem 2.3 If $q = 2$ is the number of hat colors and the number of players is $n = 2^m - 1$, with $n > m$, then there exists a perfect strategy S such that the probability of winning is $\frac{n}{1+n}$.

Proof. [1]. □

In our example above, for the perfect strategy we obtained $L_S = \{000,\ 111\}$ when $n = 2^2 - 1 = 3$. Notice that L_S is the binary Hamming code of length 3 (Ham(2,2)). In general, the set L_S for a perfect strategy S, will be identified with the binary Hamming code of the same length. For a brief introduction to Hamming codes see [6].

Theorem 2.3, along with our knowledge of binary Hamming codes (their parameters are: length $n = 2^m - 1$ for some $m < n$, order $M = 2^n$, and minimum distance $d = 3$), allow us to consider perfect strategies for larger team sizes. The next value of n for which we are guaranteed the existence of a perfect strategy occurs when $n = 2^3 - 1 = 7$. In this case our vector space of possible configurations will contain $2^7 = 128$ vectors. Once we find a perfect strategy we can then expect the team to win with a probability of $\frac{7}{8}$.

Coming up with a set of deterministic instructions, a strategy, for each player is not as immediate for $n = 7$ as it was for $n = 3$. The following theorem, however, becomes very useful:

Theorem 2.4 A strategy S is perfect for the n-player binary hat problem if and only if the set L_S, corresponding to the set of losing configurations for that strategy, is a perfect 1-covering of F_2^n.

Proof. [1]. $\square$

3. The hats game for $n = 7$ players

Referring to Theorem 2.4, since Hamming codes provide a perfect 1-covering of F_2^n, the codewords in the losing set should maintain the properties of a Hamming code. Just as in the $n = 3$ case, two codewords in the set of losing configuration should be the two vectors of F_2^n consisting of all one color. To select the other 14 members of L_s, notice that there are $\binom{7}{3} + \binom{7}{4} = 70$ different configurations each containing three of one color and four of the other. Recalling that Hamming codes have minimum distance 3, we decide to choose the other fourteen vectors from these 70. The following is one possible set of losing configurations, where red$= 0$ and blue$= 1$.

$$L_s = \begin{bmatrix} 0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1 \\ 0&0&1&1&1&0&1 \\ 1&1&1&1&0&0&0 \\ 0&1&0&1&1&1&0 \\ 0&1&1&0&0&1&1 \\ 1&0&0&1&0&1&1 \\ 1&0&1&0&1&1&0 \\ 1&1&0&0&1&0&1 \\ 1&1&0&0&0&1&0 \\ 0&0&0&0&1&1&1 \\ 1&0&1&0&0&0&1 \\ 1&0&0&1&1&0&0 \\ 0&1&1&0&1&0&0 \\ 0&1&0&1&0&0&1 \\ 0&0&1&1&0&1&0 \end{bmatrix}$$

With the players numbered from 1 to 7, each vector from F_2^7 represents a possible configuration of red (0) and blue (1) hats on the players. With these players and L_s, a perfect strategy can be developed.

Strategy: Each player is to memorize the set of losing configurations listed in L_s above. After seeing the hats of his or her six teammates, each player can construct two possible configurations by filling in 0 or 1 in his or her own position of the vector. The players are instructed to pass if they have to choose from two winning configurations or from two losing configurations. On the the other hand, if one configuration is in L_s and the other is not, they are instructed to choose the latter. Thus, when the hats actually form a losing configuration, each player will guess incorrectly since they will have a losing and winning configuration as their choices and were instructed to choose the winning configuration. If a configuration occurs that is not in L_s, then only one person will have a choice between a losing and winning configuration. Therefore, for all vectors outside of L_s , one person will guess correctly, the other players will pass, and the team will win.

Next we provide some constructions for L_S when $n \geq 7$.

Construction A

Let's take a closer look at L_S for the seven player team from the last section.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Using the middle (4^{th}) column of L_S, which will henceforth be referred to as DC,(for the *dividing column*), we can separate each row of the matrix into two vectors, both in F_2^3. The array to the left of the dividing column will be referred to as FP (for the *first part*) and the array to the right of the dividing column will be referred to as SP (for the *second part*).

$$FP = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad DC = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad SP = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Observe first that the vectors in the first eight rows of SP, are just the binary expansions of the respective rows. A repeat of these rows (in the same order) gives us rows 9 through 16. One can easily see that FP consists of all the vectors in F_2^3 (without repetition). The order in which these occur is less obvious however. To see this, consider the binary Hamming code of length 3:

$$C_0 = \begin{cases} 0 & 0 & 0 \\ 1 & 1 & 1 \end{cases}$$

First we list all the cosets $e_i + C_0$ of C_0, where $i \in \{1,\ 2,\ 3\}$ and e_i is the vector with a single '1' in the i^{th} position. The cosets of C_0 are:

$$C_1 = \begin{cases} 0 & 1 & 1 \\ 1 & 0 & 0 \end{cases} \quad C_2 = \begin{cases} 1 & 0 & 1 \\ 0 & 1 & 0 \end{cases} \quad C_3 = \begin{cases} 1 & 1 & 0 \\ 0 & 0 & 1 \end{cases}$$

So with this notation we see that FP can be written as:

$$FP = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ - & - & - \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ - \\ C_3 \\ C_2 \\ C_1 \\ C_0 \end{bmatrix}$$

With this we then have the following model for L_S when the number of players on the team is 7. Here k corresponds to the row of L_S.

$$\begin{bmatrix} C_0 & 0 & F_2^3 \\ C_1 & 0 & \\ C_2 & 0 & \\ C_3 & 0 & \\ - & - & - \\ C_3 & 1 & F_2^3 \\ C_2 & 1 & \\ C_1 & 1 & \\ C_0 & 1 & \end{bmatrix}$$

The following is another construction for L_S when $n = 7$, but it can also be used to generate perfect strategies when $n > 7$.

Construction B

Using the construction as described in [5], all perfect codes with length $n \geq 3$ can be obtained by using a perfect binary repetition code and its translates and an extended perfect code and its translates. For our purposes, the perfect code and its translates as described above will be designated by C, and the extended code and its translates will be designated by C^*. The following is an example of $n = 7$ illustrating the use of translates that leads to a theorem pertaining to upper bounds of all perfect codes.

Example : For $n = 7$, it seems logical that out of seventy different configurations there would be several different sets of sixteen vectors with a minimum distance three. Instead of trying several combinations of vectors with $min(d(C)) = 3$ to find an upper bound for perfect strategies of $n = 7$, the construction of perfect codes of length seven described above becomes very helpful.

This particular construction involves concatenating C with C^*.

The following extended perfect binary repetition code of length four and its translates are used:

$$C_0^* = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \; C_1^* = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \; C_2^* = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \; C_3^* = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Notice that all the translates of C^* are of even weight. For this particular construction, the extended perfect code is of even weight, resulting in its translates covering the space of E_2^n.

According to the construction, C and C^* can be concatenated in any form in order to obtain both linear and nonlinear perfect codes of length seven. Since C has length three, the number of ways they can occur in a length seven code is to choose any three columns out of seven. The same can be thought of with C^*, except now there are four columns to choose from out of the seven. This results in $\binom{7}{3} = \binom{7}{4} = 35$ different ways to concatenate C and C^*.

Because C has 3! different ways to be written and C^* has 4! different ways to be constructed, each codeword would have 3!4! different ways to place $0's$ and $1's$. Thus, the total number of ways to develop perfect codes of length seven is $\binom{7}{3} * 3!4! = 7!$

Theorem 3.1 A loose upper bound for the amount of perfect strategies for a code of length $n = 2^r - 1, (r \geq 3)$ is $n!$.

Proof. When n has the form $2^r - 1$, one can always use a perfect code of length n_t and its translates and an extended perfect code of length $(n+1)_t$ and its translates. Therefore, out of the n columns of a perfect code, there will be n_t columns to choose from for the perfect code and its translates and $(n+1)_t$ columns to choose from for the extended perfect code and its translates. With $n = n_t + (n+1)_t$, this results in $\binom{n_t + (n+1)_t}{n_t} = \binom{n_t + (n+1)_t}{(n+1)_t}$. This must then be multiplied by the number of ways C and C^* can be written, resulting in

$$\binom{n_t + (n+1)_t}{n_t} * n_t!(n+1)_t! = n!$$

□

We now look at the next Hamming length of $n = 15$. The losing set for $n = 15$ contains 2048 words. It is more complicated in its design but can be broken down quite nicely mimicking the $n = 7$ case:

$$\begin{pmatrix} G_0|D_0 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ G_1|D_1 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ G_2|D_2 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ G_3|D_3 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ ----- & ----- \\ G_3|D_4 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ G_2|D_5 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ G_1|D_6 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \\ G_0|D_7 & \begin{cases} \mathbf{0} & \to F_2^7 \\ \mathbf{1} & \to F_2^7 \end{cases} \end{pmatrix}$$

The matrix above has 16 *repeats*: the dividing column changes from a string of zeroes to a string of ones 16 times. The second part of the matrix is generated using all of the codewords of F_2^7. As with our previous case, a repeat has either a zero or a one on the dividing column with the binary representation of all the codewords in F_2^7 in the second part.

The first part of the matrix formed by concatenation using two sets of codewords. Each vector in G_i is of length four, and each vector in D_j is of length three, that is, G_i is a $2^{11} \times 4$ matrix and D_j is a $2^{11} \times 3$ matrix. All the rows of D_j are binary the representation of j. Hence D_0 is 000, D_1 is 001, D_2 is 010 and so on.

The following is a useful method for generating the sets G_i, but first, we will need some new terminology. A set of translates G_i is a sequence of translate leaders. A *translate leader* is a vector of F_2^4 that has even weight and is denoted by g_i. As mentioned earlier, each translate leader is an element of C_0^*, C_1^*, C_2^*, or C_3^*. Each translate leader is used to generate two sets of 16 vectors in F_2^4. Each set of translates G_i contains a different sequence of all the g_i's.

Eight codewords $\{0000, 0111, 1011, 1100, 1101, 1010, 0110, 0001\}$ denoted as $c_0, ..., c_7$ and their complements $\bar{c_0}, ..., \bar{c_7}$ are listed. The same eight codewords are listed a second time but with their order reversed. This generates a total of 32 vectors of length 4. If we denote this list by l_0, we get

$$l_0 = \begin{pmatrix} c_0 = g_0 \\ \bar{c_0} \\ \vdots \\ c_7 \\ \bar{c_7} \\ \bar{c_7} \\ c_7 \\ \vdots \\ \bar{c_0} \\ c_0 = g_0 \end{pmatrix}$$

Notice that c_i and g_i are not the same, except when $i = 0$; a c_i is a vector chosen from F_2^4 of either odd or even weight, with its complement listed right below it. A translate leader g_i, on the other hand, is a vector of even weight that is added to every codeword generated by each of the c_i's. Also,the first vector in the list always corresponds to the translate leader $c_i + C_0 = c_i$ for all i.

Hence the first translate leader g_0 is the codeword 0000. The second list of 32 vectors is generated by adding 0011 (which comes right after the end of l_0) to all the vectors l_0. Therefore 0011 is the second translate leader, denoted as g_1. Thus we obtain the second list l_1.

$$l_1 = \begin{pmatrix} c_0 = g_0 & + & 0011 \\ \bar{c_0} & + & 0011 \\ \vdots & & \\ c_7 & + & 0011 \\ \bar{c_7} & + & 0011 \\ \bar{c_7} & + & 0011 \\ c_7 & + & 0011 \\ \vdots & & \\ \bar{c_0} & + & 0011 \\ g_0 = c_0 & + & 0011 \end{pmatrix} = \begin{pmatrix} c_0 = g_0 \\ \bar{c_0} \\ \vdots \\ c_7 \\ \bar{c_7} \\ \bar{c_7} \\ c_7 \\ \vdots \\ \bar{c_0} \\ c_0 = g_0 \end{pmatrix} = l_0 + 0011$$

Thus we can generate eight different lists each time by adding a translate leader to each of the codewords in l_0. This constitutes the first set, G_0. In general, to obtain the entire list of l_i's, one only has to add a translate leader g_i to it.

$$l_i = \begin{pmatrix} c_0 = g_0 & + & g_i \\ \bar{c_0} & + & g_i \\ \vdots & & \\ c_7 & + & g_i \\ \bar{c_7} & + & g_i \\ \bar{c_7} & + & g_i \\ c_7 & + & g_i \\ \vdots & & \\ \bar{c_0} & + & g_i \\ g_0 = c_0 & + & g_i \end{pmatrix}$$

The other sets and their translate leaders are:

$$G_0 = \begin{cases} 0000 + l_0 \\ 0011 + l_0 \\ 0101 + l_0 \\ 0110 + l_0 \\ 1001 + l_0 \\ 1010 + l_0 \\ 1100 + l_0 \\ 1111 + l_0 \end{cases}, G_1 = \begin{cases} 0110 + l_0 \\ 0101 + l_0 \\ 0011 + l_0 \\ 0000 + l_0 \\ 1111 + l_0 \\ 1100 + l_0 \\ 1010 + l_0 \\ 1001 + l_0 \end{cases}, G_2 = \begin{cases} 1010 + l_0 \\ 1001 + l_0 \\ 1111 + l_0 \\ 1100 + l_0 \\ 0011 + l_0 \\ 0000 + l_0 \\ 0110 + l_0 \\ 0101 + l_0 \end{cases}, G_3 = \begin{cases} 1100 + l_0 \\ 1111 + l_0 \\ 1001 + l_0 \\ 1010 + l_0 \\ 0101 + l_0 \\ 0110 + l_0 \\ 0000 + l_0 \\ 0011 + l_0 \end{cases}$$

4. Miscellaneous conjectures about L_S

In section 3 we considered the case of perfect codes and their strategies. There we were able to observe some similarities between the cases $n = 3, 7$, and 15. Below we propose certain formulas for based only on our observations of those cases we considered.

It is our hope that any set of losing configurations can be split into a first part, a dividing column, and a second part. Hence we will make some conjectures about binary Hamming codes. The first and second parts together forms a matrix with $n = 2^{m-1} - 1$ columns. Any vector in the first part may be of the form $(G_i|D_j)$, where D_j is of perfect length and G_i is considered to be *extended perfect*, that is, of the length $n = 2^m$. For example, for $n = 15$, G_i was of length 4 while D_j was of length 3. For n=31, G_i may be of length 16, and D_j may be of length 15.

We present below some possible formulas. As mentioned earlier, $n = 2^m - 1$. $|L_m|$ is the order of the losing set of the m^{th} order ($m > 2$), R represents the number of repeats in a losing set, A is the number of columns in D_j, B is the number of columns in G_i, N is the total number of groups G_i's, and C tells us how many vectors each group G_i generates.

m	**n**	$\mathbf{\lvert L_m \rvert}$	**R**	**A**	**B**	**N**	**C**
3	7	2^4	2	1	–	–	–
4	15	2^{11}	2^4	3	2^2	2^3	2^8
5	31	2^{26}	2^{11}	7	2^3	2^7	2^{19}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	n	$2^{2^{m-1}-m}$	$\frac{2^{2^{m-1}-m}}{2^{2^{m-1}}}$	$2^{m-2}-1$	2^{m-2}	2^A	$\frac{\lvert L_m\rvert}{2^A}$

We can prove that our formula for the $|L_m|$ is correct. We already know that the size of our losing set is $\frac{1}{n+1}$.To find the size of the losing set for a general n, we only have to solve :

$$\frac{1}{n+1} = \frac{|L_m|}{2^n}.$$

If we solve for $|L_m|$, we get

$$|L_m| = \frac{2^n}{2^m}.$$

Using the fact that $n = 2m - 1$, we solve in terms of m:

$$|L_m| = 2^{2^{m-1}-m}.$$

We can also propose a recursive formula for finding the size of the m^{th} losing set, when we have a perfect number of players, $n = 2^m - 1$ players. We will take $m = 2$ case our basis. The general formula for the size of the m^{th} order of losing set is:

$$\begin{cases} |L_2| = 2 \quad \text{base case} \\ |L_m| = 2^{2^{m-1}-1} \cdot |L_{m-1}| \end{cases}$$

This formula also helps us find a recursive formula for the number of repeats R in a losing set: the number of repeats in a losing set $|L_m|$ is $|L_{m-1}|$, i.e. the number of vectors in the previous losing set. The basis for this deduction is also $|L_2| = 2$.

$$\begin{cases} |L_2| = 2 \quad \text{base case} \\ R = \frac{L_m}{2^{2^{m-1}-1}} \end{cases}$$

We can see this correlation between R and $|L_m|$ by looking at the table above.

If our assumption are correct, these formulas could give us a clearer view on the level of complexity of any given losing set. They could also be the building blocks for a more efficient enumeration of all the vectors in a losing set and lead to a standardization of the structure of all losing sets.

5. Introduction to Optimal Strategies

When the number n of players is not perfect, that is, $n \neq 2^m - 1$ for any m, the game cannot be played using a perfect strategy for all n players. However, an optimal strategy can be found. The optimal strategies found thus far have yielded the same winning probabilities of the perfect strategy of greatest length n' such that $n > n' = 2^m - 1$. [8]

One such optimal strategy is to have n' players play using the perfect strategy and the remaining players pass. [8,1] Thus, the winning configurations of length n can be constructed using a trivial direct sum construction

$$C_1 \oplus \; C_2 = \left\{ (c_1, c_2) : \; c_1 \,\epsilon\, C_1, \;\, c_2 \,\epsilon\, C_2 \right\}$$

Where C_1 is a set of losing configurations of Hamming length n', and $C_2 = F_2^{n-n'}$, and the symbol "," represents concatenation. [3]

For example, a possible set of losing configurations for $n = 4$ players is found using $C_1 = \{000, 111\}$ and $C_2 = \{0, 1\}$. Note that the first three players participate and the fourth player always passes.

$$L_s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Likewise when $n = 5$, a possible set of losing configurations is found by using $C_1 = \{000, 111\}$ and $C_2 = \{00, 01, 10, 11\}$. Again, note that the first three players participate and the fourth and fifth players always pass.

$$L_c = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We are interested in finding optimal strategies that allow the participation of all team members while winning with the same probabilities. We will refer to the subset of $n' = 2^m - 1$ players who play according to the perfect strategy as the *critical* players, and the remaining $k = n - n'$ players as the *residual* players. A residual player simply guesses her hat color to be the same as that of any critical player(s).

In the case of four players, there are three critical players, and the remaining player is a residual player. An example of a set of losing configurations for the four player strategy is the set

$$L_s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

The winning configurations differ from their losing configurations as follows:

0000	1110	1001	0111
1000	0110	0001	1111
0100	1010	1101	0011
0010	1100	1011	0101
0001	1111	1000	0110

Strategy: Note that four winning configurations $\{0001, 0110, 1000, 1111\}$ are repeated in this enumeration. These winning configurations differ from the losing configurations in positions 1 and 4. When these configurations occur, players 1 and 4 will guess their hat color correctly together. All other winning configurations differ from the losing configurations in either position 2 or 3. When these configurations occur, either player 2 or player 3 will guess correctly to win the game.

Consider the configuration 0001. Player 1 sees either 0001, a winning configuration, or 1001, a losing configuration, and she will correctly guess 0. Player 2 sees either 0001 or 0101, both of which are winning configurations, and so he passes. Player 3 sees either 0001 or 0011, both of which are winning configurations, and she also passes. Player 4 sees either 0000, a losing configuration, or 0001, a winning configuration, and he guesses 1 correctly. Notice that players 1 and 4 guess correctly together.

Consider the configuration 1100. Player one sees either 0100 or 1100, both of which are winning configurations and she passes. Player two sees either 1000 or 1100, both of which are winning configurations, and he also passes. Player three sees either 1100, a winning configuration, or 1110, a losing configuration, and so she correctly guesses 0. Player four sees either 1101 or 1100, both of which are winning configurations, and he passes.

When looking at this strategy from the geometric perspective, it is seen that the vector space F_2^4 can be covered with four spheres of radius one. Thus, the *covering radius* of the four losing configurations is 1: $cr(L_s) = 1$. However, because there is overlap of these spheres the four losing configurations do not represent a perfect packing. Therefore the *packing radius* must be 0: $pr(L_s) = 0$. These radii are related such that

$$cr(L_s) = pr(L_s) + 1.$$

We will call such strategies *quasi-perfect* corresponding to the associated quasi-perfect code.

Next we will investigate how these losing sets are constructed.

Let $\{x_1, x_2\}$ and $\{y_1, y_2\}$ represent two of the translates discussed $n = 7$ case, and let $a \epsilon F_2$.

$$L_s = \big\{(x_1, a),\ (x_2, a),\ (y_1, \overline{a}),\ (y_1, \overline{a})\big\}$$

The twelve winning configurations that are at a distance of one away from the four losing configurations can be found by changing a single digit of each losing configuration. Let v_k be the zero vector with a 1 in the k^{th} position, then for each $c \epsilon L_s$, we have a winning configuration $c + v_k$. Four winning configurations will be a distance of one away from two distinct losing configurations in unique positions i and j. When these configurations occur, players i and j will guess correctly and the other two players will pass. The eight remaining winning configurations are a distance of one away from exactly one losing configuration. When these winning configurations occur, one player, i nor j, will guess correctly while the other three players will pass. When a losing configuration occurs, all four players will guess incorrectly.

Conjecture : *Using the above construction, a loose upper bound for the total number of losing sets for* $n = 4$ *is* 48.

Using the translates of length three as in the perfect case of $n = 7$, equivalent losing sets can be developed by pairing the perfect code and its translates with one another and adding a parity such that each pair of translates has the same parity check digit added. For example, by pairing C_0 with C_1, C_2, and C_3 with the appropriate parity check digit added as a fourth column, the following six codes are obtained.

$$L_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \qquad L_2 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \qquad L_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

$$L_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \qquad L_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \qquad L_6 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Thus, for L_1 it can be seen that the first two rows consist of C_0 with a 0 added onto each row, and the last two rows consist of C_1 with a 1 added to each row.

The remaining losing sets can consist of C_1 paired with C_2 and C_3, and C_3 paired with C_4. These three pairings will produce six more losing sets with the two possible parity check columns added. Thus, by using the various pairs of the perfect binary repetition code and its translates with an appropriate parity check digit added as the fourth column, twelve different losing sets can be constructed. Since the parity check column could be used in any column, there appears to be a total of 48 different losing sets for the case of $n = 4$. This can be seen more clearly by using the combination formula to choose two pairs of translates out of the four to represent the four rows of the code, and multiplying by two for the two different ways to write a parity check column for each code. Since the parity check column can go in any column, this number is then multiplied by four to represent the parity check in each column. Therefore we have $\binom{4}{2} * 2 * 4 = 48$ possibilities.

Next we turn our attention to the case of the team of 6 players, noting that this team size is one away from perfection when $n = 7$. In their paper *"On Hats and Other Covers"* the authors describe the case of the team size $n = 2^m - 2$ as the worst case for the hats game. Finding all codewords with $min(d(C)) = 3$ to compile the losing set, one can find up to eight. These eight, with a covering radius of one, do not cover the space. As a result, there are eight vectors leftover, which also all happen to be a minimum distance of three away from each other. Thus, all eight of these vectors must be included in the losing set, resulting in sixteen codewords in the losing set. With sixteen out of sixty-four vectors in the space F_2^4, the winning probability is only $\frac{3}{4}$.

Using the definition of a *shortened Hamming code* [4], it is hoped that a strategy leading to a winning probability better than $\frac{3}{4}$ can be developed. A shortened Hamming code has length $2^m - 2$ and 2^{2^m-2-m} codewords, where $m \geq 2$. This code is a nearly perfect code obtained from the perfect Hamming code of length $2^m - 1$ by deleting all codewords ending in 0 and deleting the last column of the remaining codewords. This new code has half as many codewords as the Hamming code but still retains its covering capabilities and the $min(d(C)) = 3$ property of a Hamming code. The main difference between a shortened Hamming code and a Hamming code is that the shortened Hamming code has a covering radius of two and a packing radius of one. A Hamming code, however, has a packing and covering radius of one since it is perfect. With this difference, it seems as though a different type of strategy is required in order for the set of eight to be the only losing vectors in the space.

Although the above eight codewords cover the space with non-disjoint spheres as in the $n = 4$ and $n = 5$ cases, complications occur using the same strategy as in the case of four players. The following code is the set of losing configurations found from one of the perfect codes of $n = 7$:

$$L_s = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Normally a player would guess if he or she had a choice between a codeword in L_s and a vector in F_2^n, not in L_s. If a player had the choice between two vectors outside of L_s, then they would pass. This strategy would result in correct guesses for the winning configurations, and all of the players guessing incorrectly for each configuration in L_s. For this particular losing set, however, we find that if the zero vector is the actual hat configuration, we have a case where all players pass. For example, player one would have the choice of the zero vector or 100000, player two would choose between the zero vector and 010000, player three would have the zero vector or 001000, and so on. Since none of the players choices are listed in the code, all players would pass. In addition to the zero vector, seven other words will result in all players passing. It is interesting to note that all eight of these vectors were the ones leftover from the Hamming code of length 7 with the last column of 0's deleted.

Although it seems logical to invent a new strategy that would result in the eight codewords used to cover the space being the only losing vectors, it has now been shown that a strategy cannot be developed such that only eight vectors are in the losing set. It has been found that overlap occurs between the eight vectors leftover from the Hamming code and all other vectors that are one away from a codeword. For example, if the zero vector is the hat configuration of the team, then every player will have the choice of the zero vector, which is the complement of $111111 \in L_s$, or a vector that is one away from one other codeword. In addition, if 100000 is the hat configuration, one player will have the choice between 100000, which is one away from a codeword, and a vector that is a complement of a codeword. Thus, if players decide to guess the vector one away from a codeword when given the choice between a complement and a vector one away from a codeword, then the team will lose all configurations leftover from the Hamming code of length seven. Thus, eight more vectors will be a part of the losing set, leading back to the original conjecture of a winning probability of $\frac{3}{4}$. In [8] it is shown that $n = 6$ could have a winning probability of $\frac{6}{7}$ with a covering radius of one, leaving us further strategies to investigate. In fact, there are many questions left open about the hat problem, and the construction of perfect and imperfect codes in general. As Freeman Dyson once said, "the bottom line for mathematicians is that the architecture has to be right. In all the mathematics that I did, the essential point was to find the right architecture. It's like building a bridge. Once the main lines of the structure are right, then the details miraculously fit."

Acknowledgements

First we would like to thank the National Science Foundation, the National Security Agency, and Miami University for supporting us and the entire SUMSRI program. Next we would like to thank Dr. Dennis Davenport, Bonita Porter, Dr. Patrick Dowling, Dr. Vasant Waikar, and everyone involved with SUMSRI for their hard work and enthusiasm for the program. We also want to extend our appreciation to our professors who have prepared us for this experience. Special thanks to Dr. Tom Farmer for his help in the construction of this paper. Finally, we appreciate the guidance and support from our seminar advisor, Dr. Jillian McLeod, and our graduate assistant, Angela Grant. We could not have done this without them!

References

[1] M. Bernstein, *The Hat Problem and Hamming Codes*, Focus, Vol. **21**, No. **8** (2001), 4-6.

[2] J. Blowers, *Hats and Hangar Queens*, http:\\ jimvb.home.mindspring.com/mathematics.htm 2001.

[3] G. Cohen, I. Honkala, S. Lytsin, and A. Lobstein, *Covering Codes*, North-Holland Publishing Company, Amsterdam, 1997.

[4] W. Duckworth II, *Chapter 5: Less-Than-Perfect Codes*, Codes, Designs, and Distance, http:\\www.public.iastate.edu/ wmd/diss/chap5.pdf, 1998, 39-47.

[5] T. Etzion and A. Vardy, *On Perfect Codes and Tilings: Problems and Solutions*, SIAM J. Discrete Math, **11**, Vol **2** (1998), 205-233.

[6] D. Hankerson, D. Hoffman, D. Leonard, C. Lindner, K. Phelps, C. Rodger and J. Wall *Coding Theory and Cryptography: The Essentials*, Marcel Dekker, New York, 2000.

[7] M. Krishnamoorthy, *Lecture 9 on 09/27/01*, http:\\www.cs.rpi.edu/ moorthy/Courses/DA/Lectures/Lec9.html.

[8] H. Lenstra, G. Seroussi, *On Hats and other Covers* , Extended Summary.

[9] S. Robinson, *Why Mathematicians Now Care about Their Hat Color*, The New York Times, 4/10/2001.