From Guessing Games to Compact Discs:
Some Applications of Coding Theory*

Leslie Calloway[1] Jacquis Casher[2] Stacy Hoehn[3]

July 18, 2002

Abstract. Packings and coverings from coding theory naturally lend themselves to many applications. In our paper we first investigate the q-ary hat color problem, which makes use of both perfect packings and coverings. In particular, we look at possible losing sets and strategies for the cases where there are perfect numbers of players. Then we make modifications to the rules of the hat game to increase the winning probability and make most efficient use of the information gained from configurations in the losing set. Using our modified rules and the ternary linear Golay code, we derive a perfect strategy for a team of eleven players and three hat colors. We then leave perfect coverings and turn to strong coverings as a method to play the q-ary hat game. We also briefly discuss some applications of error-correcting codes.

1. Introduction

A codeword is a sequence of symbols from an alphabet F_q. A collection of such codewords of the same length is called a block code; a code can also be thought of as a subset of F_q^n. If q is prime, F_q is a field, and F_q^n is the vector space of all vectors of length n. A linear code is a specific type of block code whose codewords also form a vector space with the nice properties of closure under scalar multiplication and vector addition. For example, if $C = \{00000, 11110, 00001, 11111\}$, C is a linear code and 00000 is one of the codewords. An important property of codes is the Hamming distance, which is the number of positions in which two codewords differ. The codewords 00000 and 11110 have a Hamming distance of four. The minimum Hamming distance of C is one since every pair of codewords differs in at least one position [6].

Using the Hamming distance, we can interpret codes geometrically. We can construct spheres centered at the codewords to include additional vectors from the space. The vectors that lie within a sphere of radius t are those that are a distance of no more than t from a given codeword. The vector space is *packed* when these spheres have the maximal radius, called the packing radius, that still allows them to be disjoint. The vector space is *covered* when the spheres have the minimal radius, called the covering radius, that allows every vector in the space to fall within at least one sphere. A perfect code is one where the packing radius equals the covering radius, in other words, every vector in the space falls within exactly one sphere [1].

Linear perfect codes have been classified into different categories. One trivial perfect code consists of all of the vectors in the space F_q^n; in this case, spheres of radius zero are

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

[1] Alabama A&M University, 4900 Meridian Street, Normal, AL 35762 lcalloway@aamu.edu

[2] Alabama A&M University, 4900 Meridian Street, Normal, AL 35762 jcasher@aamu.edu

[3] Xavier University, 3800 Victory Parkway, Cincinnati, OH 45207 hoehnsl@xu.edu

disjoint and cover the space. Other trivial perfect codes are those that consist of only one codeword and the repetition codes of any given length. In addition, the Hamming codes are perfect codes. These codes, having alphabet F_q (q prime), are of length $n = \frac{q^m-1}{q-1}$ with $m \geq 2$, and their minimum distance is three. Spheres of radius one centered at the codewords perfectly pack and cover the entire space F_q^n. The remaining linear perfect codes are the two perfect Golay codes. The first is a ternary code of length 11 with 729 codewords that have a minimum distance of five; spheres of radius two centered at these codewords are disjoint and cover the space F_3^{11}. The other perfect Golay code is a binary code of length 23 with 2048 codewords that have a minimum distance of seven; spheres of radius three centered at these codewords are disjoint and cover the space F_2^{23}. There are also non-linear perfect codes that have the same dimensions as the Hamming codes [6].

We can use perfect codes, packings, and coverings for many different applications. One is the hat color problem, often referred to as the hat game. This game involves a host, a team of n players, and q hat colors. After the contestants are blindfolded, the host tosses a coin to determine what color hat to place on each contestant's head. Once the hats are in place, the blindfolds are removed, and the contestants are free to look at the colors of their teammates' hats. When the signal is given, all players must simultaneously guess their own hat color or else pass. However, at least one person must guess, and winning the prize of $1 million dollars requires that there be at least one correct guess and no incorrect guesses. No communication is allowed during the game, but the contestants are allowed to meet beforehand to discuss a fixed strategy [1].

Since the color of each hat is chosen independently by the toss of a coin, one might initially think a player would not gain any useful information about his hat color by looking at his teammates' hats. With this in mind, one might agree that a good strategy would be to have everyone pass except one designated player, who always guesses a particular color [1]. With this strategy, the winning probability would be $\frac{1}{q}$. Coding theorists, however, realized that applying perfect packings and coverings can dramatically increase this probability.

Every ordered arrangement of the team's hat colors is a vector of length n, called a *configuration*. To increase the probability of winning, a team develops a plan for how to guess the correct configuration. This plan, called a *strategy*, is a set of specific instructions for each player to follow once the game starts. Using a given strategy S, there will be a set of losing configurations, denoted L_S, that becomes our code. There is also a set of winning configurations, denoted W_S. A perfect strategy is one that maximizes the size of W_S and minimizes the size of L_S. Note that a perfect strategy does not guarantee that every configuration will result in a win and that perfect strategies only exist whenever L_S is a perfect code.

Much research has been done for the case $q = 2$ and $n = 2^m - 1$ with $m \geq 2$. With these parameters, a Hamming code exists, so the team can develop a set of losing configurations (which turns out to be a Hamming code if we replace hat colors by 0's and 1's) that perfectly cover the space with spheres of radius one. For any team size n,

there are 2^n total configurations, and $\binom{n}{0} + \binom{n}{1} = 1 + n$ configurations will lie in each sphere. Therefore, $\frac{2^n}{1+n}$ configurations are needed in the losing set in order to cover the entire space.

Once a team comes up with a losing set, a possible perfect strategy for this case is as follows: Look at the colors of the hats of the players around you and construct two configurations by filling in a 0 or 1 in your position. If neither of the resulting configurations are losing configurations, pass. If one of the configurations is in the losing set, guess the color option that did not yield this configuration. Using this strategy, only the $\frac{2^n}{1+n}$ losing configurations will result in losses, while the remaining $\frac{n2^n}{1+n}$ configurations will result in wins. Thus, in the cases where n is of the form $n = 2^m - 1$, the probability of winning is $\frac{n}{n+1}$, which approaches one as the number of players increases.

Less is known about the case where there are $q > 2$ hat colors. Can we use perfect coverings once again to develop strategies for this scenario? Do we need to make slight modifications to our coverings or to the game itself in order to achieve high probabilities of winning? These are a few of the questions that we will investigate.

2. Exploring the q-ary Hat Color Problem

2.1 The Original q-ary Hat Color Problem

If the hat game is played with q hat colors, where $q > 2$, can we still use perfect codes to build our strategy? First let's see what happens if three hat colors are used with four players. Since $n = \frac{q^m-1}{q-1} = \frac{3^2-1}{3-1} = 4$, we know that a Hamming code exists for this situation, with $m = 2$. To find this Hamming code, denoted $Ham(2, 3)$, we can use its parity check matrix. In general, an $n \times (n-k)$ matrix H is a called a *parity check matrix* of a code of dimension k if and only if the product of each codeword and H is the $\underline{0}$ vector. With Hamming codes, H is a $n \times m$ matrix whose rows consist of nonzero vectors in F_q^m that are not scalar multiples of each another [6]. Thus, in the case of $Ham(2, 3)$, a parity check matrix is

$$H = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

From this matrix, the code C is the set of all vectors in F_3^4 whose product with H is 00: $C = \{0000, 0111, 0222, 2102, 2210, 1012, 2021, 1120, 1201\}$. To relate this code to the hats problem, we can think of C as L_S, the set of losing configurations.

A strategy built around this losing set could be as follows: Look at the colors of the hats of the players around you and create three configurations by filling in 0, 1, and 2 in your position. If none of the resulting configurations are losing configurations, i.e. elements of L_S, pass. If one of the configurations is in L_S, guess the smallest of the remaining choices. Let's see what happens when this strategy is applied.

Case one - the actual hat configuration is in L_S:

Suppose the actual hat configuration is 0000. Looking at the colors of his teammates' hats, player 1 knows that the actual configuration is either **0**000, **1**000, or **2**000. Since **0**000 is in L_S, following the strategy, player 1 guesses the smallest of **1**000 and **2**000. Therefore, he will incorrectly guess 1. Player 2 deduces that the actual configuration is either 0**0**00, 0**1**00, or 0**2**00. Once again, 0**0**00 is the only losing configuration, so he will guess the smallest remaining choice, 1, and will be wrong. Similarly, players 3 and 4 will also incorrectly guess 1. In all situations where the actual hat configuration is a member of L_S, all four players will guess incorrectly, and the game will be lost.

Case two - the actual hat configuration is not in L_S:

Suppose the actual hat configuration is 1021. Player 1 deduces that the actual configuration is either **0**021, **1**021, or **2**021. Since **2**021 is in the losing set, by the above strategy, he will guess 0. Player 2 looks at the configurations 1**0**21, 1**1**21, and1**2**21 and notices that none of these vectors are in L_S, so he will pass. Similarly, players 3 and 4 will also pass. Only player 1 will guess, and he will be correct only 50% of the time since there are two color options remaining after the one that gives a configuration in L_S is eliminated. For example, in this case where the actual hat configuration is 1021, player 1 knows that a 2 in his position must result in a losing configuration, but he does not know whether his actual hat color is 0 or 1. In the case where the actual hat configuration is 1021, the team will lose the game, but if 0021 is the actual configuration, player 1's guess of 0 will be correct and the team will win. Thus, half of the configurations not in the losing set will lead to wins, while the other half will result in losses.

The probability of winning using this strategy can easily be determined. Out of the $3^4 = 81$ possible configurations, nine are in the losing set, which we know will never result in wins. Of the remaining 72, only 36 will result in wins. Therefore the probability of winning is $\frac{36}{81} = \frac{4}{9}$. This is an improvement on the $\frac{1}{3}$ probability of winning when using the strategy where one player guesses and he always guesses a particular color. But is there any way to further improve the odds of winning? The authors of [8] suggest that when using perfect coverings and following the rules of the original hat game, the best possible winning probability will only approach $\frac{1}{q-1}$ for large values of n.

In the above examples, all configurations not in the losing set result in three passes and one player knowing one color that he is not. If we still want to use perfect coverings, can we make slight modifications to the game that would allow the other players to make use of this information in order to determine their own hat colors?

2.2 The Modified Hat Color Problem

Modified Rules for the Hat Game

1. Players are allowed more than one round in which to guess their hat colors.
2. During each round, all players must simultaneously say "My hat color is i," "My hat color is NOT i," or "Pass," where i is one of the color options. However, if all players pass in any round, then the team immediately loses.
3. The rounds continue, with each player making a statement or passing, as long as

no incorrect statements are made.

4. The team wins and the game ends if one person correctly guesses their own hat color and no incorrect statements have been made.

Let's apply these rules to the scenario of three hat colors, four players, and L_S from above. If the actual configuration is in L_S, say 0000, everyone will still guess incorrectly when they say "My hat color is not 0". However, if the actual configuration is not in L_S, say 1021, players 2, 3 and 4 will pass, while player 1 will declare, "My hat is not 2." This is a correct statement, so the game continues to a second round. In this round, the other players now know that filling a 2 in for player 1's position must result in a losing configuration. For example, player 2 knows that 2?21 is a losing configuration. Looking at the set of such configurations, the only one of that form is 2**0**21, so player 2 now knows that he must be a 0. He declares that hat color when guessing in round two. Likewise, players 3 and 4 will now be able to correctly declare their hat colors. Player 1 will pass in round two because he does not have any additional information. All three guesses in this round will be correct, so the team will win. This is true for any configuration that is not in L_S.

The probability of winning using the modified rules is much higher than when the original rules are used. Of the 81 configurations, only the 9 losing configurations will result in losses. Thus, the probability of winning is $\frac{72}{81} = \frac{8}{9}$.

For all values of q and n corresponding to Hamming codes, i.e. $n = \frac{q^m-1}{q-1}$ for some $m \geq 2$, the modified hat game will always require at most two rounds. This is because the minimum distance between configurations in a Hamming code is three, so when players look for a losing configuration that has the appropriate hat colors for the other players, replacing the hat color of the player who guessed "My hat is not: " with the appropriate color, there will only be one configuration that meets these criteria. For configurations in the losing set, every player will guess incorrectly in the first round and the game will be over. If the team's hat configuration is not a losing configuration, one player will say what color he is not, and the rest will pass during the first round. During the second round, this player will pass while the others will correctly determine their hat colors, resulting in wins. With the modified hat game, this particular change in the rules from the original hat game leads to just enough information for the contestants to learn a great deal about their own hat colors. This version of the game makes the best use of perfect coverings in cases where $q > 2$.

For the values of q and n for which Hamming codes exist, there are $\frac{q^n}{1+n(q-1)}$ losing configurations with the modified hat game. Therefore, the probability of winning is $\frac{n(q-1)}{1+n(q-1)}$. Note that this formula generalizes the winning probability of $\frac{n}{n+1}$ when $q = 2$.

Even with restricting both the original and modified versions of our game to only $q = 3$ hat colors, the number of configurations quickly escalates. The next value of n for which there exists a perfect covering is 13. $Ham(3, 3)$ has dimension 10, so there will be $3^{10} = 59,049$ members in the losing set out of the $3^{13} = 1,594,323$ total configurations in the space. Although a team size of 13 is reasonable, it is not reasonable for them to

actually consider a code L_S of this size when formulating their strategy unless the use of computers is allowed.

More than three hat colors can be used. In the case of $q = 5$ hat colors and $n = 6$ players, a Hamming code can once again be used to determine a set of losing configurations that perfectly covers the entire space. $Ham(2, 5)$ has dimension 4, so there will be $5^4 = 625$ losing configurations, out of a total of 15,625 configurations in F_5^6. Thus, using the modified strategy, the winning probability is $\frac{24}{25}$, compared to a $\frac{8}{25}$ probability of winning using the regular rules.

2.3 Applying the Golay Code to the Hat Color Problem

Next we consider the possibility of perfection when the team size is not of the form $\frac{q^m-1}{q-1}$. When there are eleven players and three hat colors, we cannot employ a Hamming code since $11 \neq \frac{3^m-1}{3-1}$ for any m. However, there does exist another perfect code, the Golay 11 code, denoted $\mathcal{G}_{11}$, that proves useful. $\mathcal{G}_{11}$ was introduced by Marcel J. E. Golay during the 1940s as a perfect double-error correcting ternary code with 729 codewords of length 11 and minimum distance five [6]. A natural question is whether or not $\mathcal{G}_{11}$ can be used as a set of losing configurations even though this perfect code has a covering radius of two, meaning that every member of W_S is at most two away from a member of $L_S = \mathcal{G}_{11}$. Is there a perfect strategy that corresponds to this perfect code?

Possible Strategy Using the Modified Rules:

In the first round, compare the configuration of hats that you can see to the list of configurations in $\mathcal{G}_{11}$. If, ignoring your position, you are within one of some configuration in $\mathcal{G}_{11}$ and the color corresponding to your position in that configuration is i, state "My hat is not i." Otherwise, pass. In the second round, proceed normally in determining your correct hat color.

Any configuration of hat colors is either zero, one, or two away from a member of $\mathcal{G}_{11}$. Using the above strategy, all configurations in $\mathcal{G}_{11}$ will result in losses, as each player notices that they are within one of that configuration and states "My hat is not _ ." That is, each player will guess incorrectly in each of these cases in the first round, and the game will be over.

Any configuration that is one away from a configuration in $\mathcal{G}_{11}$ will also result in a loss. In the first round, there will be ten incorrect guesses and only one correct guess, so the game will be lost. For example, suppose 22222222222 is in $\mathcal{G}_{11}$ and the actual configuration is 22222022222. Player 1 will see ?2222022222, will determine that 22222222222 is the nearest codeword in $\mathcal{G}_{11}$, and will incorrectly say "My hat is not 2." Players 2 through 5 and 7 through 11 will similarly be incorrect. Player 6 will also say "My hat is not 2," but this player will be correct. Since there are incorrect statements, however, the game will be still be lost.

Wins come from the configurations that are two away from codewords in $\mathcal{G}_{11}$, corresponding to its perfect covering and packing radius. For example, if the actual configuration is 22220022222, depending on a player's position, it may appear to be two away

from 22222222222 and/or 22220020200 in $\mathcal{G}_{11}$. In round one, since players 1 through 4 and 7 through 11 notice that they are two away from one of these configurations, following the strategy they will pass. Players 5 and 6 will see a configuration that is one away from 22222222222, and they will both say "My hat is not 2." Both will be correct. In the second round, every other player will now be able to determine his hat color. Since no incorrect guesses will have been made, the game is won in these cases.

Although this strategy works in theory, it is not a very practical approach to the problem. In order to carry out the strategy, a player will need to compare three possible configurations to each of the 729 losing configurations before making a decision on how to guess. An easier way to find the nearest losing configuration is to make use of the *syndrome* of each of the three possible configurations. The syndrome of a configuration $\underline{v}$ from F_q^n is the product of $\underline{v}$ and a parity check matrix H. The syndrome is the linear combination of the rows of H which correspond to the positions in which $\underline{v}$ differs from the nearest losing configuration. If the syndrome of $\underline{v}$ is the zero vector, then $\underline{v}$ is a losing configuration. If the syndrome of $\underline{v}$ is either a row of H or a scalar multiple of a row, then $\underline{v}$ differs from a losing configuration in only one position. If the syndrome is any other vector, then $\underline{v}$ differs from a losing configuration in two or more positions. With this new information, we can rewrite the previous strategy in a way that is more useful for a player when he is guessing his hat color.

Revised Strategy (still using the modified rules):

In the first round, create the vectors $\underline{v}_1, \underline{v}_2$, and $\underline{v}_3$ corresponding to the configurations that result from filling 0, 1, and 2 in your position. Then compute the three possible syndromes by multiplying $\underline{v}_1, \underline{v}_2$, and $\underline{v}_3$, respectively, by a parity check matrix for $\mathcal{G}_{11}$,

$$H = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 0 & 2 & 1 & 1 & 2 \\ 2 & 0 & 2 & 1 & 1 \\ 1 & 2 & 0 & 2 & 1 \\ 1 & 1 & 2 & 0 & 2 \\ 2 & 1 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

If the syndrome is 00000 when the color x is substituted in for your position, state "My hat is not x." Otherwise, if exactly one of the syndromes is a nonzero multiple of a row of H, state "My hat is not y," where y is the color that gave you that syndrome. In all other cases, pass. In the second round, proceed as normal in determining your hat color.

This strategy is equivalent to the original strategy without the exhausting number of comparisons. Both versions of this strategy for the modified hat problem with eleven players and three colors result in losses whenever the actual configuration is within a Hamming distance of one of a member of $\mathcal{G}_{11}$. Twenty-three configurations are within one

of each member of $\mathcal{G}_{11}$. Since there are 729 different elements in this set, $23 * 729 = 16,767$ out of the $3^{11} = 177{,}147$ possible configurations will be losing configurations. The other 160,380 configurations will be winning. Therefore, the probability of winning is 0.9054, which is very high considering there are three hat colors to choose from.

Still we ask the following question: Is it possible to come up with a strategy that has an even higher probability of winning? Is it possible to develop a strategy that also results in wins for the configurations within one of a codeword?

2.4 The Original Game and Strong Coverings

Can we increase the probability of winning the hat game when $q > 2$ without changing the rules? In [8], it is suggested that is impossible to do so with perfect coverings. Other types of coverings, called *strong coverings*, however, have been found to work well for this purpose when the value of n would have produced a perfect covering in the binary case.

A set C is a *strong covering* of the space F_q^n if every configuration *not* in C has one coordinate position such that when that position is replaced by every other member of the field F_q, each new configuration is in C [8].

Here is one way to come up with strong coverings for cases when the team size is perfect in the binary case. Let H_m be the parity check matrix of $Ham(m, 2)$, and let $n = 2^m - 1$. Then a strong covering C of the space is the set of configurations in F_q^n whose syndrome contains no zeros [8].

For example, if $q = 3$ and $n = 2^2 - 1 = 3$, then

$$H_m = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

C is the set of configurations from F_3^3 whose syndromes do not contain zeros, i.e. their syndromes are 11, 12, 21, or 22. Thus,

$$C = \{100, 220, 110, 200, 011, 101, 021, 111, 222, 012, 202, 022\}.$$

Any configuration not in this set has a coordinate which, when changed to the other two options in F_3, results in vectors belonging to C. For example, the third coordinate of 020 can be changed to a 1 or 2, and both resulting configurations are in C. Since this is true for any configuration not in C, we say that C is a strong covering for the space F_3^3.

Once a strong covering is created, the original rules of the hat game can be used, and wins will occur for all configurations not in C. If two of the configurations resulting from the substitution of 0, 1, and 2 in your position are in C, guess the one remaining option. Otherwise pass. If the actual configuration is in C, all players will guess incorrectly, therefore losing the game. Otherwise, one player will correctly guess and the rest will pass, thereby winning the game.

For example (using the strong covering from above), if the actual configuration is 100, player 1 will see that 100 and 200 are in C, so he will incorrectly guess "My hat is 0."

Player 2 will see that 100 and 110 are in C, so he will incorrectly guess "My hat is 2." Player 3 will see that 100 and 101 are in C, so he will incorrectly guess "My hat is 2." Since there are incorrect guesses, the team loses.

If the actual configuration is 112, however, the team will win. Player 1 will notice that only 012 is in C and will pass. Player 2 will see that none of his vector options are in C and will also pass. For player 3, the configurations 110 and 111 will be in C, so he will correctly guess "My hat is 2." No incorrect guesses are made in the process, so the game is won.

For $q = 3$ and $n = 3$, there are 15 configurations that are not in C and will result in wins. Therefore, the probability of winning will be 15 out of the total of 27, or $\frac{5}{9}$.

The next value of n for which this process can be used to determine a strong covering is $n = 2^3 - 1 = 7$. A parity check matrix is

$$H_m = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The strong covering C consists of all of the configurations in F_3^7 whose syndromes are 111, 112, 121, 211, 122, 212, 221, or 222. This set has 648 configurations out of the 2187 in the space. The other 1539 are winning configurations. For each element of C, i.e. the losing configurations, each player will guess incorrectly. For each of the other configurations, six players will pass and one player will correctly guess his hat color. Therefore the probability of winning is $\frac{1539}{2187} = \frac{19}{27}$.

We are interested in constructing strong coverings for values of n that are not of the form $2^m - 1$. We first consider the case $n = 4$. One strong covering is

$$C = \begin{cases} 0100, 1100, 2100, 0011, 1011, 2011 \\ 0222, 1222, 2222, 0220, 1220, 2220 \\ 0101, 1101, 2101, 0012, 1012, 2012 \\ 0110, 1110, 2110, 0021, 1021, 2021 \\ 0202, 1202, 2202, 0200, 1200, 2200 \\ 0111, 1111, 2111, 0022, 1022, 2022 \end{cases}.$$

Every configuration not in this set has one coordinate position such that when every other member of the field F_3 is placed in that position, each new configuration is in C. This strong covering is essentially the same as the $n = 3$ case for players 2 through 4, ignoring the color of player 1's hat. Of the 81 possible configurations in the space, 36 are losing and 45 are winning. Therefore, the probability of winning is once again $\frac{45}{81} = \frac{5}{9}$, which is still much better than the $\frac{1}{3}$ winning probability using the naive strategy or the $\frac{4}{9}$ winning probability using a perfect covering with the original rules. However, this winning

probability is much lower than the winning probability of $\frac{8}{9}$ using the modified rules of the hat game.

3. History and Applications of Coding Theory

Throughout our investigation we have repeatedly used techniques and notions developed by Richard Hamming. Thus we will include a brief historical account of his life. In 1945, Richard Hamming joined the Manhattan project. He became well-known for his pioneer work in error-correcting and error-detecting codes. Hamming won the 1996 Eduard Rheim Award for his work in this area. Hamming realized that there was a way to use as few bits as possible and still receive the correct message, but he was unable to explicitly prove this [10].

Hamming's work piqued the interest of other mathematicians, such as Claude Shannon who worked with the information theory aspects of coding to achieve clear information transmission. Some of Shannon's work provides us with the great sound quality of compact discs. Even though compact discs may have visible scratches and thumb prints, a compact disc player still reads the song accurately. This is because of the error-correcting capabilities built into the compact discs.

The basic units of a compact disc are arranged in 32 bit words (codewords). There are 4 billion possible codewords in the space containing these words, but we only take a fraction of those to form our actual code. We make sure the codewords in the newly formed configuration are as different as possible because we do not want any two codewords in the configuration to be the same distance away from an incorrect word. This code is used to encode the music. When an error occurs, the compact disc player reads the received vector as the codeword it is closest to in the list of configurations. In coding theory, this method of decoding is called maximum likelihood decoding.

Since four billion vectors can be read per second on a compact disc, the error-correction process must be efficient. Selecting the correct codewords from large configurations for every error is not time efficient. This process can surpass the compact disc's functional capabilities if there are too many errors in the received vector. The Reed-Solomon code is a code which corrects the error in the vector itself to create a quicker and more efficient way to correct codes. Thanks to the Reed-Solomon code we have error correcting in compact discs as long as the errors do not exceed the error-correcting capability. Now, a compact disc player can correct errors with up to 1000 consecutive bits of information [9].

There are several other areas of technology that coding theory is applied to, such as telecommunications, and of course there are also many computer applications. Sphere packing research can even be applied to molecular biology. The enzyme EcoRI precisely travels along deoxyribonucleic acid (DNA) and finally binds to the nucleotide sequence 5'GAATTC3'. The nucleotide sequence is composed of the bases guanine, adenine, thymine, and cytosine. In the molecular process of denaturation, the enzyme EcoRI precisely slices the DNA to create complementary bases with binding capabilities. This process occurs when the EcoRI is in a high-energy state. In the low-energy state of hybridization the molecules lose energy and anneal together using the complementary bases again to

re-form the GAATTC sequence. If the molecule does not lose any energy it will travel past the specified nucleotide sequence. The EcoRI stays there until it repeats the slicing process again. This replication process is important to the evolution of living systems. This process of denaturation and hybridization forms what is called a *molecular machine*. A molecular machine is a large spherical macromolecule that needs energy to function.

A molecular machine can exist in two states, the "after" state and the "before" state. The before state is the state when the machine has not begun operating. This corresponds to the state in communications systems when a codeword has not yet been sent. In the after state the molecular machine has performed its tasks and dissipated energy. In terms of communication, the after state is the state when the decoder receives the symbol (codeword) from the message, exhausting the message symbol. Therefore, each sphere represents one possible message and one possible molecular state.

In the language of coding theory, denaturing or heating the molecular machines causes the EcoRI to expand. The sphere that results in this high energy state is thought of as the outer sphere. After the machine has performed its operation, the machine loses energy and the sphere contracts. We can relate the different energy states of the molecular machine to a gumball machine. The larger outer sphere represents the molecular machine at high energy while the smaller spheres represent all the possible lower energy states. Hence, sphere (gumball) packing by molecular machines is analogous to how an array of codewords are structured. With this analogy, it becomes clear that the molecular machine in its varying energy states represent a sphere packing problem. If we think of the smaller spheres as the codewords and the outer sphere as the code, we can see that an optimal scenario is one in which there is no overlap of the gumballs.

Further, the number of gumballs equates with the number of codewords for a code. We are also able to parallel molecular machine capacity, the number of gumballs that we can pack in the gumball machine, with channel capacity in coding theory. In information theory, the channel capacity is $\log_2 \|C\|$. If we consider the channel capacity as the number of possible symbols that can comprise a bit, and molecular machine capacity as the number of possible states the machine can choose between, then they are functionally the same.

All molecular machines encounter thermal noise. For example, when an enzyme methyl alters the base pairs in the nucleotide sequence, the enzyme EcoRI is no longer able to attach itself to the GAATTC sequence in DNA. Sometimes, the methyl group does even not attach itself to the appropriate GAATTC sequence; it may attach to a sequence one base away such as: CAATTC. This corresponds to a single-error in the attachment process [12].

4. Further Questions

We still have many questions to answer regarding some of the applications of coding theory. For example, with the q-ary hat color problem, what other strategies will result in high probabilities of winning? Also, what kinds of strategies work well with the modified rules when the team size is not perfect? When using strong coverings, what is the minimum size of these coverings when the team size is not perfect? What strategy would allow us to

apply the other perfect Golay code, $\mathcal{G}_{23}$, to the game? Would this strategy be practical for players to carry out during the game? Overall, we are curious about the countless other areas in which we suspect coding theory can be applied.

5. Acknowledgements

This research would not have been possible without the help of many people and organizations. Thanks to the National Science Foundation, the National Security Agency, and Miami University for supporting the SUMSRI program. We would also like to thank Dr. Dennis Davenport, Bonita Porter, Dr. Patrick Dowling, Dr. Vasant Waikar, and everyone involved with SUMSRI for their contributions to the program. Also we extend out thanks to Dr. Tom Farmer for helping with this paper. Finally, we appreciate the guidance we received from our seminar advisor, Dr. Jillian McLeod, and our graduate assistant, Angela Grant. We cannot thank everyone enough for their help!

6. 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] R. Moody, *Why Curiosity Driven Research?*, http://www.math.mun.ca/ edgar/moody.html.

[10] J. O'Connor, E. Robertson, *Richard Hamming*, http://www-history.mcs.st-andrews.ac.uk/history/Mathematicians/Hamming.html.

[11] S. Robinson, *Why Mathematicians Now Care about Their Hat Color*, The New York Times (Apr 2001), http://www.msri.org/activities/jir/sarar/010410NYTArticle.html.

[12] T. Schneider, *Examples of Molecular Machines*, http://www.lecb.ncifcrf.gov/ toms/paper/ccmm/latex/node2.html.