A Genetic Algorithm for Designing Constellations with Low Error Floors

Matthew C. Valenti and Raghu Doppalapudi
West Virginia University
Morgantown, WV
Email: {mvalenti,doppala}@csee.wvu.edu

Don Torrieri
U.S. Army Research Lab
Adelphi, MD
Email: dtorr@arl.army.mil

***Abstract*—The error floor of bit-interleaved coded modulation with iterative decoding (BICM-ID) can be minimized for a particular constellation by maximizing the harmonic mean of the squared-Euclidian distances of signals whose labels differ in just one bit position. This problem has been formulated as an instance of the Quadratic Assignment Problem (QAP) and solved using the Reactive Tabu Search (RTS). In this paper, we propose a genetic algorithm for solving the symbol labeling problem and show that it usually yields designs that are isomorphic to those obtained using RTS. We then extend the algorithm to optimize not only the labelings of the signal points, but also their location in the signal space. Using this approach, new constellations of cardinality 16, 32, and 64 are evolved that have an error floor lower than that of QAM or PSK modulation. The new constellations exhibit gains of 1.32 and 0.88 dB over the best 16-QAM and 64-QAM constellations reported previously.**

I. INTRODUCTION

Bit-interleaved coded-modulation (BICM) is a standard approach for coding in modern wireless systems involving a binary encoder followed by a bit interleaver and a nonbinary modulator [1]. Relative to trellis-coded modulation (TCM), BICM increases diversity by replacing a symbol interleaver with a bit interleaver, and hence has improved performance over fading channels. Due to its simple implementation and superior performance over fading channels, BICM has become a standard feature in all proposed future-generation cellular, satellite, and wireless networking systems.

Performance in a BICM system can be improved by allowing the demodulator and decoder to iteratively exchange extrinsic information. Such systems are known as BICM with iterative decoding and demodulation (BICM-ID) [2]. Plots of the bit error rate for BICM-ID systems generally exhibit a waterfall region, which is characterized by a rapid decrease in the bit error rate as the signal-to-noise ratio increases, and an error-floor region, in which the bit error rate decreases much more slowly. For a given signal set, the performance in both regions is determined by the labeling map, which specifies the $\log_2(M)$ bit sequence associated with each of the M symbols in the signal set. In this paper, we are concerned with designing constellations and labeling maps that reduce the error floor.

Methods for generating good labeling maps that provide low error floors for a given signal set have been previously described in [3]–[5]. In [4] it is shown that the minimization of the so-called *error-free feedback* (EFF) bound, which is an approximation of the actual error floor, can be formulated as an instance of the combinatorial optimization problem known as the *quadratic assignment problem* (QAP) [6]. In particular, the goal is to maximize the harmonic mean of the Euclidian distance of signals whose labels differ in just one bit position. The optimization is done over the $M!$ possible labeling maps. In [4], *Reactive Tabu Search* (RTS) [7] was used to perform this optimization. However, the optimization can be solved in other ways, and in this paper we propose a genetic algorithm for optimizing the labeling map of an arbitrary signal set.

The error floor depends not only on the labeling map, but also on the location of the signal points. Previous work has only focused on the design of the labeling map and assumed a fixed modulation type, typically QAM or PSK. However, the error floor can be reduced further by moving the constellation points. We modify the genetic algorithm to design not only the labeling map, but also the placement of the points in the constellation. By evolving the design across multiple generations, new modulation formats are found with error floors that are lower than previously known results involving QAM or PSK.

The remainder of the paper is organized as follows. Section II provides a model of the system under consideration, while Section III derives the error-free feedback bound and identifies the main contributors to the bound. Section IV formulates the problem of minimizing the EFF bound for a given constellation as a combinatorial optimization problem, and Section V proposes a genetic algorithm for solving the problem. Section VI modifies the genetic algorithm to allow the points in the constellation to be moved. Numerical results showing the EFF bounds for optimized QAM, PSK, and evolved constellations with $M = \{16, 32, 64\}$ are given in Section VII and compared against the simulated error rates of actual BICM-ID systems. Finally, Section VIII concludes the paper.

II. SYSTEM MODEL

A BICM-ID system is shown in Fig. 1. In the transmitter, a vector $\mathbf{u}$ of message bits is passed through a rate r_c binary convolutional encoder to produce a codeword $\mathbf{c}'$. The codeword is bitwise interleaved by a permutation matrix $\mathbf{\Pi}$ to produce the bit-interleaved codeword $\mathbf{c} = \mathbf{c}'\mathbf{\Pi}$. The bit-interleaved codeword is then passed through a modulator to produce the vector of complex symbols $\mathbf{s} \in \mathcal{X}^L$ where $\mathcal{X}$

is a constellation of symbols with cardinality M and average energy $\mathcal{E}_s$. The overall rate of the system is $R = mr_c$, where $m = \log_2 M$, and the average energy per information bit is $\mathcal{E}_b = \mathcal{E}_s/R$.

The modulator selects symbols from $\mathcal{X}$ based on the constellation labeling map and the corresponding group of m consecutive bits at the modulator input. Each signal in $\mathcal{X}$ is labeled with a unique m-bit sequence $\mathbf{b}$. Define the integer representation of a length-m binary vector $\mathbf{b}$ as

$$q(\mathbf{b}) = \sum_{k=0}^{m-1} b_k 2^{m-k-1}. \quad (1)$$

Let $\mathbf{x} = [x_0, x_1, ..., x_{M-1}]$ be a vector containing each signal in $\mathcal{X}$. The signals in vector $\mathbf{x}$ are indexed according to a natural mapping so that the label $\mathbf{b}$ of x_i satisfies $q(\mathbf{b}) = i$. Let $\mathbf{x}'$ be an interleaved version of the vector $\mathbf{x}$ such that $\mathbf{x}'_k = \mathbf{x}_{\mu(k)}$ where $\mu(k)$ is a permutation function. Let $\boldsymbol{\mu} = [\mu(0), \mu(1), ..., \mu(M-1)]$ be a vector containing the integers 0 through $M-1$ permuted according to the function $\mu(k)$. When the input to the modulator is $\mathbf{b}$, then the modulator selects symbol $s = x'_{q(\mathbf{b})} = x_{\mu(q(\mathbf{b}))}$ for transmission. Thus, the vector $\boldsymbol{\mu}$ specifies the labeling map for a particular ordered signal set $\mathbf{x}$.

Each coded symbol passes through a frequency-nonselective channel with complex-fading amplitude c. In this paper, we assume uncorrelated Rayleigh fading such that the c's are i.i.d. zero-mean complex Gaussian with unit power. The output of the channel is $y = cs + n$ where n is a sample of a white complex-Gaussian process with variance $N_0/2$ per dimension.

At the receiver, a demapper within the demodulator processes each received symbol to produce a vector $\mathbf{z}$ of bit likelihoods. This vector provides extrinsic information that is deinterleaved and passed to the decoder. The soft-output decoder produces extrinisic information that is interleaved and provided to the demapper as a vector $\mathbf{v}$ of *a priori* information.

The output of the demodulator for bit k is [8]

$$z_k = \log \frac{\sum_{x' \in \mathcal{X}_k^{(1)}} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \exp\left[\beta_j(x') v_j\right]}{\sum_{x' \in \mathcal{X}_k^{(0)}} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \exp\left[\beta_j(x') v_j\right]} \quad (2)$$

where the function $\beta_j(x')$ returns the j^{th} bit of the label of x' and $\mathcal{X}_k^{(b)}$ is the set of all symbols in $\mathcal{X}$ labeled with $b_k = b$.

III. Criteria for Low Error Floors

The union bound on the bit error probability is [1]

$$P_b \leq \frac{1}{k_c} \sum_{d=d_f}^{\infty} W_I(d) f_p\left(d, \frac{\mathcal{E}_s}{N_0}\right) \quad (3)$$

where d_f is the minimum distance of the convolutional code, $W_I(d)$ is the total input weight of error events in the convolutional code at Hamming distance d, and $f_p(d, \mathcal{E}_s/N_0)$ is

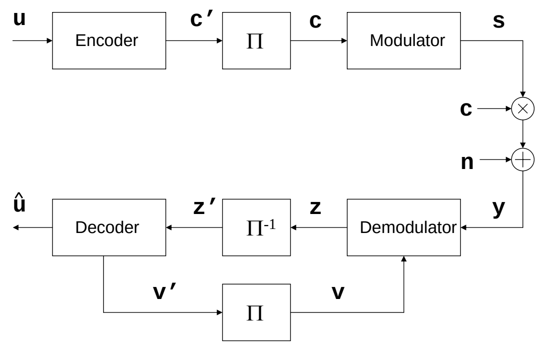

Fig. 1. Model of a BICM-ID system. Π indicates a bitwise interleaver.

the pairwise error probability (PEP). The PEP depends on the choice of constellation $\mathcal{X}$ and its labeling map $\boldsymbol{\mu}$.

Let the function $g_k(x)$ return the symbol whose label is identical to the label of symbol x except for the k^{th} position, which is complemented. The signal x and $g_k(x)$ form an *error-free feedback (EFF) signal set*. In the following, $\{x, g_k(x)\}$ is called an *EFF pair* and $g_k(x)$ is said to be the *EFF companion* of x. The *EFF bound* on BER is found from (3) under the assumption that the demodulator is provided with perfect *a priori* information by the decoder, in which case the LLR of the unknown bit k has the form

$$z_k = \log p(y|x') - \log p(y|g_k(x')) \quad (4)$$

where x' is one of the 2^{m-1} symbols whose k^{th} position is labeled with a one, i.e. $\beta_k(x') = 1$. The decision rule for bit k involves just one EFF pair for that bit, and the identity of the pair depends on the *a priori* information fed back from the decoder. Because the decoder does not always provide perfect *a priori* information to the demodulator, and because (3) is an upper bound, the EFF bound is not actually a bound but rather is an approximation. However, the EFF bound is usually able to accurately predict the performance at high SNR.

Consider how to determine $f_p(1, \mathcal{E}_s/N_0)$, i.e. the probability of a solitary bit error. Assume without loss of generality that a bit $b = 1$ is transmitted. An error will occur if the corresponding LLR z is negative

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = Pr[z < 0 | b = 1] \quad (5)$$

Since there are m bits labeling each symbol, the PEP is found by averaging the m bit error probabilities

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = \frac{1}{m} \sum_{k=0}^{m-1} Pr[z_k < 0 | b_k = 1] \quad (6)$$

The error probability for each bit may be found by averaging over the 2^{m-1} symbols that are labeled with $b_k = 1$

$$Pr[z_k > 0 | b_k = 1] = \frac{1}{2^{m-1}} \sum_{x' \in \mathcal{X}_k^{(1)}} Pr[z_k < 0 | x']. \quad (7)$$

Let $P_{z_k|x'}(z)$ denote the conditional CDF of z_k given that symbol x' was sent. Substituting (7) into (6) allows the PEP to be expressed as

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = \frac{1}{m2^{m-1}} \sum_{k=0}^{m-1} \sum_{x' \in \mathcal{X}_k^{(1)}} P_{z_k|x'}(0). \quad (8)$$

The PEP can be found using moment generating function techniques [1], [9]. Define $\phi_{z_k|x'}(s) = \mathcal{L}\left\{p_{z_k|x'}(z)\right\}$ to be the Laplace transform of the pdf of z_k given that symbol x' was sent. In Rayleigh fading [10],

$$\phi_{z_k|x'}(s) = \frac{1}{1 + s(1 - sN_0)||x' - g_k(x')||^2}. \quad (9)$$

From the integration property of the Laplace transform and the fact that the CDF is the integral of the pdf

$$P_{z_k|x'}(z) = \mathcal{L}^{-1}\left\{\frac{\phi_{z_k|x'}(s)}{s}\right\} \quad (10)$$

The PEP can thus be found by substituting (10) into (8), resulting in

$$f_p\left(1, \frac{\mathcal{E}_s}{N_0}\right) = \mathcal{L}^{-1}\left\{\frac{\psi(s)}{s}\right\}_{z=0} \quad (11)$$

where

$$\psi(s) = \frac{1}{m2^{m-1}} \sum_{k=0}^{m-1} \sum_{x' \in \mathcal{X}_k^{(1)}} \phi_{z_k|x'}(s). \quad (12)$$

When $d > 1$, the PEP is the probability that the sum of d independent bit LLRs is less than zero given that the corresponding transmitted bits are all ones. Replacing z_k in the first line of (6) with this sum and conditioning on the event that all bits are ones results in

$$f_p\left(d, \frac{\mathcal{E}_s}{N_0}\right) = Pr\left[\sum_{i=0}^{d-1} z_i < 0 \middle| \prod_{i=0}^{d-1} b_i = 1\right] \quad (13)$$

where i is used to index bit error events, which will generally be in different symbols. By using the convolution property of the Laplace transform, the PEP may be found using

$$f_p\left(d, \frac{\mathcal{E}_s}{N_0}\right) = \mathcal{L}^{-1}\left\{\frac{[\psi(s)]^d}{s}\right\}_{z=0} \quad (14)$$

Once the PEP is found for each d, the the bit error probability is found by substituting into (3).

Asymptotically, as $N_0 \to \infty$, the BER can be approximated by [1], [9]

$$P_b \approx \kappa\left[d_h^2 \frac{\mathcal{E}_s}{N_0}\right]^{-d_f} \quad (15)$$

where κ is a constant and

$$d_h^2 = \left[\frac{1}{m2^{m-1}} \sum_{k=0}^{m-1} \sum_{x' \in \mathcal{X}_k^{(1)}} ||x' - g_k(x')||^{-2}\right]^{-1} \quad (16)$$

is the *harmonic mean* of the squared-Euclidian distances between signals in each EFF signal set.

Taking the base-10 logarithm of (15), defining $X_{dB} = 10\log_{10} X$, and recalling that $\mathcal{E}_s = R\mathcal{E}_b$ yields

$$\log_{10} P_b \approx \frac{-d_f}{10}\left[\left(Rd_h^2\right)_{dB} + \left(\frac{\mathcal{E}_b}{N_0}\right)_{dB}\right] + \kappa_{dB}. \quad (17)$$

A plot of $\log_{10} P_b$ versus $(\mathcal{E}_b/N_0)_{dB}$ will be a straight line with negative slope given by d_f, which is the diversity gain. The horizontal offset of the curve is controlled by d_h^2, with larger values of d_h^2 translating into lower error rates.

From (16), it is clear that the choice of signal set $\mathcal{X}$ and symbol labeling map $\boldsymbol{\mu}$ influences the harmonic mean, and from (17) the harmonic mean determines the offset of the BER curve. Thus, to minimize the EFF bound for a given outer code, it is necessary to maximize d_h^2. For a given $\mathcal{X}$, the maximization is performed over the set of all possible $\boldsymbol{\mu}$. Such maximization is discussed in Sections IV and V. If one is free to chose not only $\boldsymbol{\mu}$ but also the signal set $\mathcal{X}$, then the maximization is over both functions, as described in Section VI.

IV. Label Mapping Optimization

Maximizing the harmonic mean given by (16) over all $\boldsymbol{\mu}$ is equivalent to finding the minimum of a cost function:

$$\min_{\boldsymbol{\mu}} \sum_{k=0}^{m-1} \sum_{x \in \mathcal{X}_k^{(1)}} ||x - g_k(x)||^{-2}. \quad (18)$$

As discussed in [4], the minimization given by (18) can be formulated as an instance of the *Quadradic Assignment Problem* (QAP) [6]. Define the *flow matrix* $\mathbf{F}$ such that element $f_{i,j} = 1$ if the labels of x_i and x_j are different in just one bit position. Since the vector $\mathbf{x}$ is indexed according to a natural labeling, $f_{i,j} = 1$ whenever the binary expansion of the integers i and j have a Hamming distance of one. Otherwise, $f_{i,j} = 0$. Define the *distance matrix* $\mathbf{D}$ with elements

$$d_{i,j} = \begin{cases} ||x_i - x_j||^{-2}, & i \neq j \\ 0, & i = j \end{cases} \quad (19)$$

For a given $\mathbf{F}$ and $\mathbf{D}$, the cost function in (18) may be found as

$$\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f_{i,j} d_{i,j}. \quad (20)$$

The previous expression is the cost function when a natural mapping is applied. If another mapping is applied by permuting $\mathbf{x}$ to obtain $\mathbf{x}'$, then a new distance matrix $\mathbf{D}'$ must be found. However, since $x'_k = x_{\mu(k)}$, $\mathbf{D}'$ is merely a column-and-row permuted version of $\mathbf{D}$. Thus, the cost function under mapping $\boldsymbol{\mu}$ can be expressed as [4]

$$\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f_{i,j} d_{\mu(i),\mu(j)}. \quad (21)$$

The optimization is to minimize the above expression with respect to all possible mapping functions (permutations) $\boldsymbol{\mu}$.

The QAP is NP-hard and may be solved in a variety of ways [6]. Exact solutions can be found using dynamic programming, cutting plane techniques, and branch and bound procedures. For $M \geq 32$, such techniques are generally not tractable. For larger problems, heuristic search methods have been considered, including RTS [7], *genetic algorithms* [11]–[13], and *simulated annealing* [14]. Conventional genetic algorithms [11] are not competitive with RTS in terms of accuracy and efficiency. However, with recent modifications [13], genetic algorithms have emerged as a very promising candidate for solving the QAP. In [4], the problem of maximizing d_h^2 was solved using RTS. In this paper, our goal is to maximize d_h^2 by using a genetic algorithm.

V. A Genetic Algorithm for Mapping Optimization

The genetic algorithm used to optimize the mapping is based on the algorithm proposed in [13]. The algorithm is initialized with a population of N randomly drawn mapping functions $\boldsymbol{\mu}_i, 0 \leq i \leq N-1$. The mapping functions are indexed in increasing order of cost so that mapping $\boldsymbol{\mu}_0$ is the best. Whenever the population changes, it is reindexed.

Breeding requires the selection of two parents $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_k$. The first parent is selected at random from the entire population except for the best ($1 \leq j \leq N-1$), while the second is selected at random from those mappings that are better than the first ($0 \leq k \leq j$). This *preferential parent selection rule* was not contemplated in [11]–[13] and improves the rate of convergence.

The two parents produce two children $\tilde{\boldsymbol{\mu}}_j$ and $\tilde{\boldsymbol{\mu}}_k$. The children are initially empty vectors of length M. A child's *direct* parent is the parent with the same index, while its *indirect* parent is the parent with a different index. The children are generated by randomly selecting λ *crossover points*, where λ is a fixed number for all breedings. The crossover points indicate the positions of each child that are inherited from its indirect parent. At the crossover points, values from $\boldsymbol{\mu}_k$ are copied into $\tilde{\boldsymbol{\mu}}_j$ and values from $\boldsymbol{\mu}_j$ are copied into $\tilde{\boldsymbol{\mu}}_k$. The remaining positions of a child $\tilde{\boldsymbol{\mu}}_\ell$ are inherited from its direct parent $\boldsymbol{\mu}_\ell$, where $\ell = \{j, k\}$. Whenever possible, values of parent $\boldsymbol{\mu}_\ell$ that are not already in its child $\tilde{\boldsymbol{\mu}}_\ell$ are copied into $\tilde{\boldsymbol{\mu}}_\ell$ at the same position. The remaining values that could not be copied from parent to child into the same position are copied into the closest open position.

The cost of each of the two children are evaluated. If a child $\tilde{\boldsymbol{\mu}}_\ell$ has a lower cost than its parent $\boldsymbol{\mu}_\ell$, then the parent is replaced with the child. If the child is not better than the parent, then an optional *mutation* process is started. A mutant is generated from $\tilde{\boldsymbol{\mu}}_\ell$ by exchanging the values stored in randomly chosen pairs of positions in $\tilde{\boldsymbol{\mu}}_\ell$. The *mutation rate* ρ is the probability that any given position in $\tilde{\boldsymbol{\mu}}_\ell$ will become exchanged with some other randomly chosen position. During the mutation process, a total of L mutants are created and the cost of each is evaluated. If any of the mutants have a lower cost than parent $\boldsymbol{\mu}_\ell$, then the parent is replaced with the lowest cost mutant.

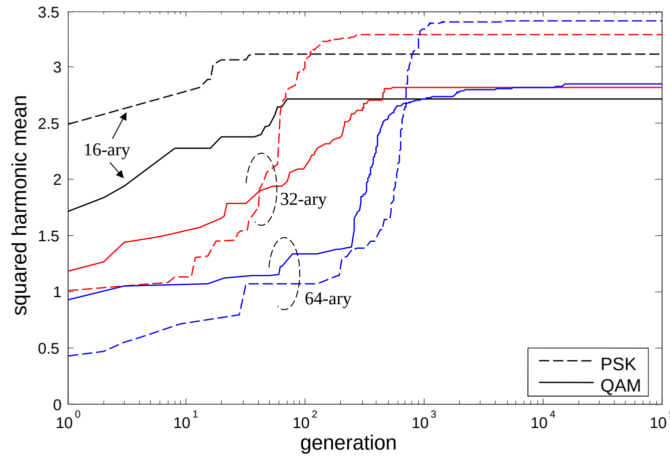

Fig. 2. Harmonic mean of the squared-Euclidian distance between EFF signal pairs after each generation of the genetic algorithm.

During each generation, a certain number of breedings occur. While this number is arbitrary, we have set the number of breedings per generation to equal N, the overall population size. As proposed in [13], our algorithm periodically performs a *culling* operation once every T_c generations. During a culling generation, each child $\tilde{\boldsymbol{\mu}}_\ell$ is no longer compared against its direct parent. Instead, the child is compared against the current worst mapping $\boldsymbol{\mu}_{N-1}$ and if the child is better than the worst mapping, it will replace the worst mapping. It is possible that the population produced by this genetic algorithm will contain duplicates of the same mapping. During culling generations, these duplicates are removed and replaced with new random mappings.

The genetic algorithm was run to optimize the mappings of PSK and QAM with $M = \{16, 32, 64\}$. For each case, the population was $N = 100$, the number of crossover points $\lambda = \lceil M/5 \rceil$, the mutation rate $\rho = 0.02$, and the culling period $T_c = M$. The number of mutants L per child was variable. Initially, $L = 0$. If after T_c generations a new best mapping is not identified, then L is incremented up to a maximum of 20. On the other hand, if a new mapping is identified within T_c generations, then L is decremented. The value of d_h^2 after each generation, up to generation 4×10^4, is shown in Fig. 2. The final value of d_h^2 along with the generation that the best mapping was identified is listed in Table I. Also shown are the values of d_h^2 obtained using RTS. As can be seen, the genetic algorithm has found the same optimal value as was found using RTS with the exception of 64-QAM, in which case the genetic algorithm's result is slightly inferior to that of RTS. The constellation labeling maps found by the genetic algorithm for $M = 16$ are shown in Fig. 3.

VI. Constellation Optimization

In the previous section, the constellation $\mathcal{X}$ was fixed and only the labeling map $\boldsymbol{\mu}$ was optimized. However, d_h^2 depends not only on the labeling map but also on the constellation. We

TABLE I
VALUE OF d_h^2 OBTAINED USING RTS AND THE PROPOSED GENETIC ALGORITHM. ALSO LISTED IS THE NUMBER OF GENERATIONS REQUIRED FOR THE GA TO CONVERGE.

M	Modulation	d_h^2 from RTS	d_h^2 from GA	generations
16	QAM	2.7190	2.7190	103
	PSK	3.1142	3.1142	1,249
32	QAM	2.8154	2.8154	542
	PSK	3.2916	3.2916	530
64	QAM	2.8742	2.8460	15,421
	PSK	3.4102	3.4102	8,987

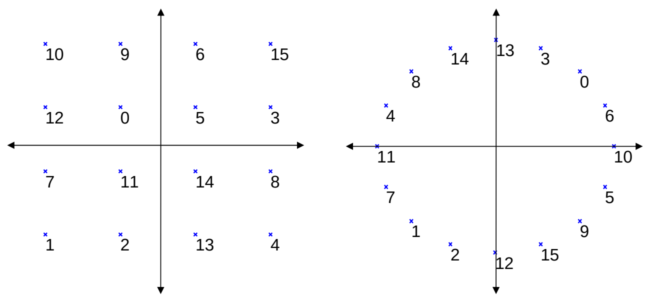

Fig. 3. Constellation labeling maps that maximize d_h^2 found using the genetic algorithm. 16-QAM is on the left, and 16-PSK is on the right.

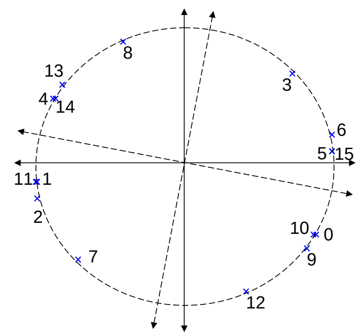

Fig. 4. 16-ary constellation optimized using a genetic algorithm. Tilted axes have been superimposed to emphasize the symmetry that has evolved.

TABLE II
SIGNAL VALUES FOR THE CONSTELLATION SHOWN IN FIG. 4.

Labeling	Real	Imaginary	Labeling	Real	Imaginary
0	0.8803	-0.4859	8	-0.4123	0.8977
1	-0.9894	-0.1141	9	0.8254	-0.5905
2	-0.9885	-0.2297	10	0.8704	-0.4840
3	0.7245	0.6716	11	-0.9993	-0.1120
4	-0.8704	0.4840	12	0.4123	-0.8977
5	0.9907	0.1081	13	-0.8254	0.5905
6	0.9885	0.2297	14	-0.8803	0.4859
7	-0.7160	-0.6677	15	0.9894	0.1141

seek to increase d_h^2 beyond the values possible for QAM and PSK by moving signal points in the constellation. Ideally, the goal is to perform the following optimization

$$\min_{\mathcal{X},\boldsymbol{\mu}} \sum_{k=0}^{m-1} \sum_{x' \in \mathcal{X}_k^{(1)}} ||x' - g_k(x')||^{-2}. \quad (22)$$

Because the number of possible $\mathcal{X}$ is infinite, this optimization is more challenging than the one given by (18).

We propose a heuristic method for performing the optimization in (22). The optimization begins with a standard constellation. Because PSK has a better potential d_h^2 than QAM, it is a more appropriate starting constellation. Several generations of the genetic algorithm are run on the PSK constellation to obtain an initial mapping. Next, the constellation is modified in an attempt to increase d_h^2. This can be done by first picking an EFF pair whose distance is is at the minimum d_e, where

$$d_e \quad = \quad \min_{\substack{x' \in \mathcal{X}_k^{(1)} \\ 0 \le k \le m-1}} ||x' - g_k(x')||. \quad (23)$$

If there are multiple EFF pairs that are distance d_e apart, then a pair is selected at random. The two points are then forced to be further apart. Let $\alpha > 1$ represent a scale factor, such that the selected EFF pair is forced to be distance αd_e apart with the same centroid. When this adjustment is made, the average energy of the constellation will typically increase, and so it must be renormalized to $\mathcal{E}_s$. Also, after the pair is pushed apart, the previously chosen labeling map might no longer be optimal, and thus it is redesigned by running one generation of the genetic algorithm with an initial population of mappings set to the final population of the previously optimized constellation. The process continues iteratively, with each iteration consisting of moving one pair of points apart, renormalizing the constellation, and genetically reoptimizing the constellation labeling map.

Starting with the 16-PSK constellation shown in Fig. 3, the optimization was run with 1000 iterations and $\alpha = 1.01$. The resulting constellation is shown in Fig. 4. The optimization resulted in $d_h^2 = 3.6841$, which is considerably larger than the best d_h^2 found with QAM and PSK. The signal values for the constellation are listed in Table II. Inspecting the constellation reveals an interesting geometry. Tilted axes have been superimposed on the figure to emphasize the symmetry in the evolved constellation. Each quadrant of the tilted constellation contains four symbols whose labels differ in exactly two bit positions; thus, there are no EFF pairs in the same quadrant. Of the four symbols in each quadrant, the first two symbols are nearly colocated, a third symbol is distance ~ 0.12 from the colocated pair, and a fourth symbol is distance ~ 0.62 from the colocated pair. The fourth symbol's EFF companions all lie on the opposite quadrant (for instance the EFF companions of '8' are $\{0, 9, 10, 12\}$). For all other symbols, two of their EFF companions are in one quadrant and the other two EFF companions are in another quadrant.

Constellations for $M = 32$ and $M = 64$ were optimized in a similar fashion. The resulting constellations had $d_h^2 = 3.5477$ for $M = 16$ and $d_h^2 = 3.5178$ for $M = 64$ which are again better than the values found for PSK and QAM. Note that for the genetically-designed constellation, d_h^2 actually decreases with increasing M, which is opposite the trend observed for QAM and PSK modulation. Thus the relative gains achieved by evolving the constellation decrease with increasing M.

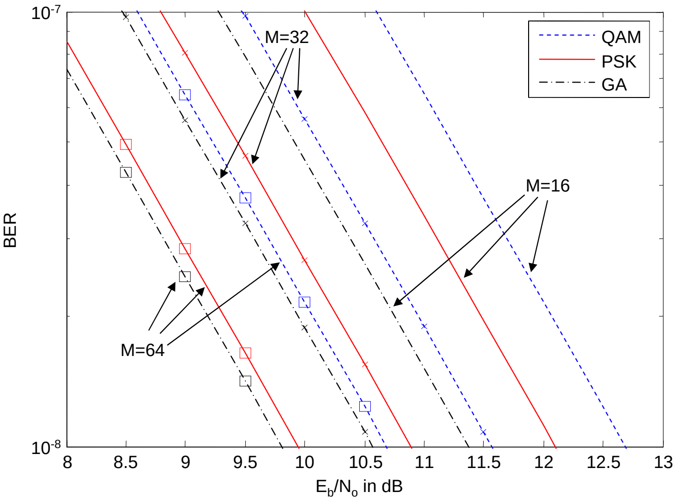

Fig. 5. The EFF bound on BER when using a $(7,5)$ convolutional code. For each M, QAM has the highest error floor and the constellation designed with the genetic algorithm (GA) has the lowest error floor.

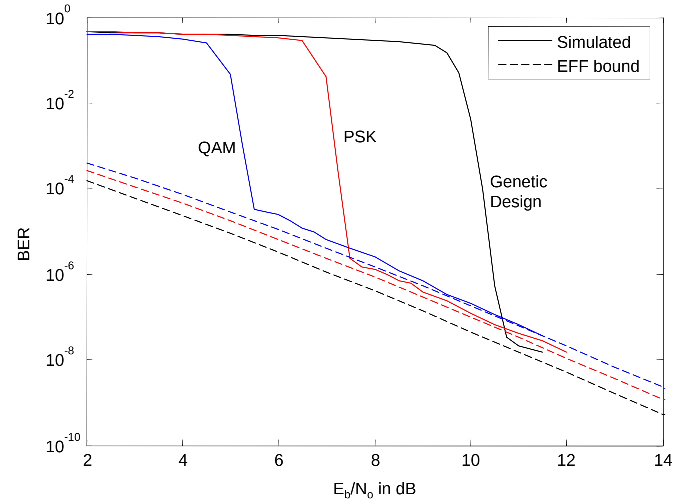

Fig. 6. Simulated bit error performance using 16-ary modulation, a $(7,5)$ covolutional code, and $24,000$-bit codewords.

VII. Numerical Results

The EFF bound was calculated for the PSK and QAM constellations described in Section V as well as the genetically-designed constellations described in Section VI. The bounds are shown in Fig. 5 for the rate-$1/2$ convolutional code with octal generators $(7,5)$ and $d_f = 5$. While all curves have the same slope due to the common value of d_f, there are gains in moving from one modulation choice to another. In general, there is a gain in moving to a larger alphabet and there are gains in moving from QAM to PSK and from PSK to the genetically-designed constellation. For instance, the genetically-designed constellations provide gains of 1.32 and 0.88 dB, respectively, over the 16-QAM and 64-QAM constellations. These trends are the opposite of the trends observed with uncoded modulation or BICM without feedback, indicating that BICM-ID achieves its gains in a much different manner. Indeed, the constellation shown in Fig. 4 would perform quite poorly without coding and iterative decoding.

To demonstrate the tightness of the EFF bound at high SNR, the three 16-ary BICM-ID systems were simulated. The simulations used a codeword length of $N = 24,000$ bits over an uncorrelated Rayleigh fading channel. The results of the simulation are shown in Fig. 6. Also shown on the plot are the corresponding EFF bounds. As can be seen, the EFF bound accurately predicts the performance at high SNR. However, it can also be seen that lowering the EFF bound comes at a cost. The waterfall region, i.e. the range of SNR characterized by a rapid drop in BER, occurs at a higher SNR for each modulation that has a lower EFF bound.

VIII. Conclusions

Evolutionary computing may be used to not only optimize the labeling map of a constellation, but also to optimize the placement of the points in the constellation. By applying an appropriate genetic algorithm, a new constellation may be evolved for BICM-ID with a predicted error floor that is on the order of 1 dB better than the best known QAM constellations. The lower error floor of the evolved constellation comes at the cost of the waterfall region being shifted to a higher SNR.

References

[1] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.

[2] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Letters*, vol. 1, pp. 169–171, Nov. 1997.

[3] J. Tan and G. L. Stüber, "Analysis and design of symbol mappers for iteratively decoded BICM," *IEEE Trans. Wireless Comm.*, vol. 4, pp. 662 – 672, Mar. 2005.

[4] Y. Huang and J. A. Ritcey, "Optimal constellation labeling for iteratively decoded bit-interleaved space-time coded modulation," *IEEE Trans. Inform. Theory*, vol. 51, pp. 1865–1871, May 2005.

[5] D. Torrieri and M. C. Valenti, "Constellation labeling maps for low error floors," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Orlando, FL, Oct. 2007.

[6] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Press, 1998.

[7] R. Battiti and G. Tecchiolli, "The reactive tabu search," *ORSA J. Comp.*, vol. 6, pp. 126–140, 1994.

[8] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded M-ary noncoherent orthogonal modulation," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 1738–1747, Sept. 2005.

[9] A. Chindapol and J. A. Ritcey, "Design, analysis, and performance evaluation of BICM-ID with square QAM constellations in Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 944–957, May 2001.

[10] S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Applications*. Kluwer, 1999.

[11] C. Fleurent and J. Ferland, "Genetic hybrids for the quadradic assignment problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 173–187, 1994.

[12] Z. Drezner, "A new genetic algorithm for the quadratic assignment problem," *INFORMS J. Computing*, vol. 15, pp. 320–330, 2002.

[13] Y. Wu and P. Ji, "Solving the quadratic assignment problems by a genetic algorithm with a new replacement strategy," *Int. J. Humanities and Social Sciences*, vol. 1, no. 3, pp. 151–155, 2007.

[14] D. T. Conolly, "An improved annealing mechanism for the QAP," *European J. of Operational Research*, vol. 46, pp. 93–100, 1990.