

# MOEA Design of Robust Digital Symbol Sets

Richard O. Day<sup>†</sup>, Abel S. Nunez<sup>‡</sup> and Gary B. Lamont<sup>†</sup>

<sup>†</sup>Department of Electrical Engineering  
Graduate School of Engineering & Management  
Air Force Institute of Technology  
WPAFB (Dayton) OH, 45433, USA

<sup>‡</sup>Air Force Research Laboratory  
Electronic Warfare Technology Branch  
Wright-Patterson, Air Force Base  
Dayton OH, 45433, USA

## ABSTRACT

Optimal constellation design is important in military digital communications for Quadrature Amplitude Modulation (QAM). Optimization realizes a reduced probability of bit error ( $P_b$ ) while keeping the same bandwidth and power for transmitting the signal. Constellation shapes currently used in QAM include rectangle, triangle, hexagonal, and concentric circles. In this study, two dimensional (4, 8, ..., and 256)-ary constellations at specific normalized signal to noise ratios ( $\frac{E_b}{N_o}$ ) having lower  $P_b$  are sought. Various models are designed to provide a Multi-objective Evolutionary Algorithm (MOEA) with a near exact model to utilize as a fitness function. MOEA found solutions are tested for merit using a Monte Carlo simulator. Comparisons of  $\frac{E_b}{N_o}$  vs  $P_b$  between the rectangular constellation and new designs are illustrated. New designs are shown to be different and comparable to the standard constellations used today

## Categories and Subject Descriptors

J.6 [Comp-aided Engr]: [Comp-aided design]

## General Terms

Algorithms, Design

## Keywords

Digital Symbol Set Design, multiobjective Evolutionary Alg

## 1. INTRODUCTION

Bandwidth efficient modulation techniques using bounded bandwidth is sought in digital communications. Symbol set design is the minimizing of the probability of bit error ( $P_b$ ) at a specific normalized signal-to-noise ratio ( $\frac{E_b}{N_o}$ ) [5]. Optimal constellations with lower  $P_b$  at specified  $\frac{E_b}{N_o}$  have numerous applications in digital communications. Lowering the inter-symbol interference reduces  $P_b$ , but normally this comes at cost of increased signal power or decreased noise interjection. Unfortunately, typical links have distortion elements

in channel filters and amplifier nonlinearities that cannot be eliminated or, in some cases, reduced. Military communications face all types of noise interjection when an enemy attempts to reduce allied information exchange. Therefore, optimizing symbol set constellations is absolutely necessary for lowering this inter-symbol interference.

Investigated<sup>1</sup> is the ability of a Multiobjective Evolutionary Algorithm (MOEA), more specifically, the multiobjective fast messy GA (MOMGA-IIa) to optimize two dimensional (4, 8, ..., and 256)-ary symbol set design for a decreased  $P_b$  at certain ( $\frac{E_b}{N_o}$ ). The combinatorics of this problem call for a stochastic search algorithm that can be used in optimizing both single- and multi-objective problems because the models generated to capture this symbol set design problem are constructed as both single and multi-criteria problems.

*Military Importance:* A well designed symbol separation ensures that a system will have a consistent performance and operating range without demanding excessive transmitter power. Symbol set design having these qualities are highly desirable for military operations especially in deployment situations when transmission power may be limited and channel noise distribution is variable<sup>2</sup>.

## 2. MULTIOBJECTIVE ALGORITHM

The MOMGA-IIa is a multiobjective version of the fmGA that has the ability to achieve a semi-partitioned search in both the genotype and phenotype domains during execution. It is an algorithm that exploits *good* building blocks (BBs) in solving optimization problems. These explicit BBs represent *good* information in the form of partial strings that can be combined to obtain better solutions. The MOMGA-IIa algorithm executes in three phases: Initialization, Building Block Filtering, and Juxtapositional.

Beginning with the Probabilistically Complete Initialization (PCI) Phase, the algorithm randomly generates a user specified number of population members. These population members are constructed to a specified chromosome length and each is evaluated to determine its respective fitness values. Our implementation utilizes a binary scheme in which each bit is represented with either a 0 or 1.

The Building Block Filtering (BBF) Phase follows by ran-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25–29, 2005, Washington, DC, USA.  
Copyright 2005 ACM 1-59593-097-3/05/0006 ...\$5.00.

<sup>1</sup>The views expressed in this article are those of the authors and do not reflect the official policy of the United States Air Force, Department of Defense, or the U.S. Government.

<sup>2</sup>This research should evolve into a dynamic symbol set system - adapting to changing channel noise distribution (usually caused by enemy signal jamming equipment) on the fly.

domly deleting loci and their corresponding allele values in each of the population member's chromosomes. This process completes once the length of the population member's chromosomes have been reduced to a predetermined BB size. These reduced chromosomes are referred to as underspecified<sup>3</sup> population members. In order to evaluate population members that have become underspecified, competitive templates (CTs) are utilized to fill in the missing allele values. Evaluation consists of the partial string being overlaid onto a CT just prior to evaluation. CTs are fully specified chromosomes that evolve as the algorithm executes. CT replacement is done after each BB generation. In the MOMGA-IIa, *competent* CTs that partition both the phenotype and genotype are selected for advancement. This innovative balance is achieved through two mechanisms: Orthogonal CT generation and Target Vector (TV) guidance. Orthogonal CT generation is used to partition the genotype space, while TV guidance is used, when MOMGA-IIa is running in multiobjective mode, to partition the phenotype space. TVs are normalized fitness markers that capture one solution per vector for future CT replacement. In the MOMGA-IIa, target vectors are used in a manner to divide the normalized fitness space of pareto-front members and select a distribution of CTs that fall nearest to each TV. Also, an orthogonal bank of chromosomes is used to filter a randomly selected CT through for creation of a set of orthogonal CTs.

The BBF process is alternated with a selection mechanism to keep only the *best* BBs found, or those with the *best* number of fitness values. In the case of a tie, where two strings each have an equal number of better fitness values (i.e. each have  $\frac{m}{2}$  *best* fitness values), the string is randomly selected between the two. It should be noted that the MOMGA-IIa has a more complex selection mechanism than MOMGA-II because it maintains more fitness values per solution. In the MOMGA-II each string has  $m$  fitness values, while in MOMGA-IIa each string has  $f = (c * m + i + o) * m$  fitness values associated with it – corresponding to the  $m$  objective functions to optimize,  $c$  competitive templates,  $i$  inverse templates (equal to  $c * m$ ), and  $o$  orthogonal templates.

Finally, the juxtapositional phase uses the BBs found in the BBF phase and recombination operators to create chromosomes that are fully specified. The MOMGA-IIa has an outer and inner loop that must be completely iterated through using each BB size and epoch before terminating. Next, the problem domain is described in detail.

### 3. FITNESS FUNCTION

Previous researchers either use different coding methods or symbol positioning to get better  $P_b$  at particular  $\frac{E_b}{N_o}$  [1, 3, 5]. In our models we allow the MOMGA-IIa to optimize both of these at the same time by assigning each symbol its bit-wise representation while assigning it a location in space. However, the challenge was found not to be what to optimize or how to represent the problem, but what model (fitness function) best represented the Monte Carlo simulation. Normally, new constellations would be tested using a Monte Carlo simulator where a random stream of symbols are encoded into a signal,  $s$ , using the amplitude and phase of symbols identified in a designed constellation. Next, noise is added to the generated signal,  $(s + n)$ , to simulate the

transmission process. The amount of noise added to the signal is related to the  $\frac{E_b}{N_o}$  under test. Then, the signal is decoded and a reconstructed symbol stream is generated from the noisy signal. Finally, the number of bit errors are calculated by comparing the reconstructed symbol stream to the originally transmitted symbol stream. This test is re-run until enough data is collected to assign a  $P_b$  rate to that particular constellation at the  $\frac{E_b}{N_o}$  under test. Four different analytical models are designed in search of the optimal balance between computational time and model correctness. Five models are tested. The first four designs are analytical approximates for this digital system and the fifth design is the Monte Carlo simulation.

Each model uses symbols placed inside a unit circle for amplitude and phase characteristics for each symbol [2]. Each symbol bit pattern is defined to be in one and only one place within the genome. Placement is in binary order,  $\{00, 01, 10, 11\}$ , and each symbol has  $d = 2$  degrees of freedom to define its location in the space.

The analytical models are simple and fast compared to the simulation model for they are designed for minimal number of calculations while having a close approximation to the real model. These fitness functions have many pre-calculated measures to help speed compute time. For all of these models, each symbol is defined in binary fashion and placed into a bit-matrix. Furthermore, each symbol's hamming code distance from one another is calculated and placed into a separate  $d \times d$  matrix called  $\mathcal{H}$ . The hamming code distance matrix keeps track of the amount of bit errors made when reconstructing the noisy signal and an incorrect symbol is selected. A complete constellation fitness computation for each and every fitness function begins with a polar to cartesian coordinates conversion on the entire symbol set. Then, a distance calculation on each symbol,  $i$ , to each other symbol,  $j$  follows. Symbol distance calculations are stored in a distance matrix  $\mathbf{D}_{i,j}$ . Once the calculations listed here are finished, each model's remaining calculations differ slightly.

**Brute force model:** The first model is called the brute force model. It takes into account intuition about how to represent this problem using a high level understanding of the problem domain. A maximization constellation fitness function is sought; thus, a negative exponential decay of the distance is used to emphasize a larger space between symbols is better. The exponential decay is modified using a constant  $K$  value to increase or decrease the rate of decay depending on the number of symbols in the constellation. Finally, to account for the bit error increasing when encountering a high hamming code distance between symbols, the exponential decaying distance is multiplied by the hamming code distance, found in the  $\mathcal{H}$  matrix, of the two symbols.

This model worked well for finding solutions that competed with the standard rectangular constellations. However, we thought accounting for symbol energy would be an improvement over this fitness function. The new fitness function describes the same fitness calculation as before, but the symbol set energy is used as a damping (dividing) factor – where energy is the sum of the squared radii for each symbol. Results of the modified brute force model were not as good as expected, resulting in development of the Volumes and False Alarm Rate models.

**Volumes model:** The volumes model is based on the principle that the  $P_b$  landscape changes with respect to each symbol. Therefore, an  $m$ -objective model is designed to ac-

<sup>3</sup>An underspecified chromosome is chromosome where some, but not all locus positions have an associated allele value.

count for each new landscape that occurs when inspecting the transmission and reconstruction of each symbol with respect to every other symbol. A hamming code distance scaled Gaussian distribution is placed on each symbol (i.e. the higher the hamming code distance, the larger the gaussian) to represent each symbol's error footprint. Next, a grid is placed on the unit circle and at each grid position the maximum value found with respect to all Gaussian distributions are summed for a fitness value. Unfortunately, this model also did not prove to be a good representation of the Monte Carlo simulation. Moreover, hundreds of solutions were found along the pareto front so determination of good solutions still required a simulation evaluation.

**False Alarm Rate model:** The false alarm rate model is designed with the premise that each symbol has the same probability of selection, but the error associated with selection changes. Furthermore, the error found using a minimum selection formula should be used as the fitness of a constellation. This model is similar to the volumes problem in that it is an  $m$ -objective problem, unscaled Gaussian distributions are placed on each symbol, a grid is placed on the unit circle and the probability of error when symbol  $m$  is transmitted and symbol  $k$  is reconstructed at grid position  $(i, j)$  is calculated for fitness.

**Minimize probability of error:** This model is designed to include the  $P_b$ , overall constellation energy ( $E_s$ ) and the  $\frac{E_b}{N_o}$  parameter into the same fitness function. Much like the Volume model, we are calculating the error associated with selecting the wrong symbol and we are assuming that the noise added to the incoming signal has a Gaussian distribution. To sum the error associated with selecting the wrong symbol, it is advantageous to find the distance of the point where the two distributions (one on each symbol location) intersect. If the two symbols have the same distribution, the same scale for the distribution, and the same standard deviation, the intersecting point is the mid point between the location of the two symbols. However, if the distributions are scaled differently, but still maintain the same standard deviation and type of distribution, there is an adjustment that can be made to find the new intersecting point.

Lastly, given that we have the same distributions and the same standard deviations, the probability of false alarm can be calculated by summing up the tail of the Gaussian away from this intersection point. This model can also be a multiobjective model - objectives are in the form of the parameter  $\frac{E_b}{N_o}$  requiring optimization (i.e. for a 2 bit constellation  $\frac{E_b}{N_o} = \{0, 1, \dots, 8\}$ , making for a 9 objectives). The relationship of  $\frac{E_b}{N_o}$  to the standard deviation,  $\sigma$ . Thus, for each objective, the Gaussian distributions changes shape as the  $\frac{E_b}{N_o}$  changes - making the  $P_b$  different for each different  $\frac{E_b}{N_o}$  even when the constellation remains the same. Finally, the power calculation is added. Unfortunately, this model also demonstrated to not be as good as the brute force model.

**Simulation model:** The simulation model acts as a correlation receiver described in the Detection of Signals in Gaussian Noise section of [4]. The number of symbols in a constellation is  $2^{S_{bits}}$ . The simulation generates  $(10000 * \log_2(\text{symbols}))$  bits randomly. These random bits are converted to 10000 symbols. The 10000 symbols are then converted to their respective coordinates in the constellation. These coordinates are then multiplied by the basis functions and the results are transmitted. Next a specific amount of

Gaussian noise is added to match the required  $\frac{E_b}{N_o}$  for a particular evaluation. The noise and signal is then correlated with the basis functions. Based on the output of the correlators, symbols are estimated according to their distances from each symbol in the constellation. Although this particular model showed to be more accurate, it is much more computationally prohibitive.

## 4. RESULTS, ANALYSIS & FUTURE WORK

MOMGA-IIa constellations having 2, 3, 4, and 5 bits all compete well with the  $x$ -rectangular constellation theoretical best. Unfortunately, comparisons with only rectangular constellations having 4, 16, 64, and 256 symbols can be made due to the limitation of the theoretical best equation. 2, 3, 6, 7, and 8 bit constellation comparisons are not illustrated here, but the only MOMGA-IIa found constellation that did not come within 1 dB of the rectangular constellation theoretical best is the 8 bit or 256ary constellation.

The MOMGA-IIa has found good constellations using the brute force model. Although MOMGA-IIa solutions did not beat the  $x$ -rectangular constellation theoretical best, they did compete rather well. This is validated using standard Amplitude Modulation techniques in use today. Theoretically, there exists a constellation that have lower  $P_b$  than the  $x$ -rectangular constellations at certain  $\frac{E_b}{N_o}$ s; however, those constellations evade researchers in the field today and the MOMGA-IIa so far.

Since, this is a new angle on an old problem. Many different techniques can be used to increase the effectiveness of the MOMGA-IIa in solving this problem. Right now, the model needs to be modified to yield maximum fitness value at the same constellation that the Monte Carlo simulation validates to be the lowest  $P_b$ . So far, a better model escapes us. Our current models show little correlation to the current simulation model. Once a model is found for finding good constellations for use on signals subject to additive White Gaussian Noise, a new model can be derived to find constellations for noisy signals having other types of noise. In fact, the ultimate goal would be to have a channel noise probe identify the noise over a channel and back propagates this information to an EA that searches for an optimal symbol set for that channel. Once the symbol set was defined for the unknown noise in the channel, communicators would have the best symbol set for that digital system.

## 5. REFERENCES

- [1] V. W.-K. Cheng and W. E. Stark. Performance of trellis-coded direct-sequence spread spectrum with noncoherent reception in a fading environment. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 47(2):579–590, May 1998.
- [2] T. W. Dittmer. Digitally modulated rf system. Primer, Harris Corporation, Quincy, IL USA, 1999.
- [3] L. L. Joiner and J. J. Komo. Improved performance of reduced m-ary orthogonal signaling using reed-solomon codes. *MILCOM*, 02/6(0-7803-5538-5/99):4, 1999.
- [4] B. Sklar. *Digital Communications*, volume I. Prentice Hall, Engle Cliffs, New Jersey, 1st edition, 1988.
- [5] C. M. Thomas, M. Y. Weidner, and S. H. Durrani. Digital amplitude-phase keying with m-ary alphabets. *IEEE Transactions on Communications*, Com-22(2):pg 168–180, Feb 1974.