Exponential Sums With Genetic Algorithm

Bobby Orozco (Mentor: Dr. Riley Casper)

California State University Fullerton

Abstract

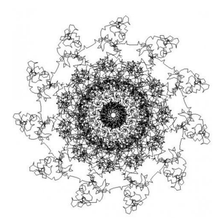
We explore a problem in exponential sums using a genetic algorithms & the role of field automorphisms. Exponential sums count solutions to Diophantine equations, which are connected to Fermat's Last Theorem, & elliptic curve cryptography.

Reverse Engineering Exponential Sums

Exponential sums are sums of the form

$$\sum_{j=0}^{N-1} e^{2\pi i f(j)/N},$$

where f is a real-valued function. These kinds of sums generate stunning images, like the one below [1].



Graph of the exponential sum $\sum_{n=1}^{161} e^{2\pi i\sin(11+n^2/21+n^3/31)}$

For integer-valued f, the sum counts solutions of diophantine equations modulo N:

$$\sum_{k=0}^{N-1} \#\{0 \le j < N : f(j) = k\} e^{2\pi i k/N}.$$

Example: If $f(j) = j^2$ and $N = 13$, then f has the following values

j	0	1	2	3	4	5	6	7	8	9	10	11	12
$f(j) \mod N$	0	1	4	9	3	12	10	10	12	3	9	4	1

and therefore

$$\sum_{j=0}^{N-1} e^{2\pi i f(j)/13} = 1+2e^{2\pi i/13}+2e^{6\pi i/13}+2e^{8\pi i/13}+2e^{18\pi i/13}+2e^{20\pi i/13}+2e^{24\pi i/13}$$

This shows us that $j^2 = 7$ has no solutions and $j^2 = 3$ has two solutions.

Our problem: In this project, we want to go backwards. Given the final value of an exponential sum, as a single complex number

$$a + ib$$

we want to find integers $a_0, a_1, \ldots, a_{N-1}$ such that

$$a + ib = \sum_{k=0}^{N-1} a_k e^{2\pi i k/N}.$$

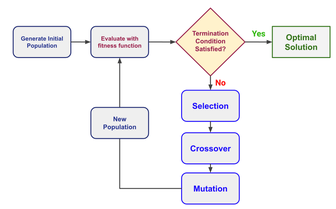
For example, what exponential sum results in the final value $-0.5 + 0.24078731i$? The answer is

$$e^{2\pi i/7} + e^{6\pi i/7} + e^{10\pi i/7},$$

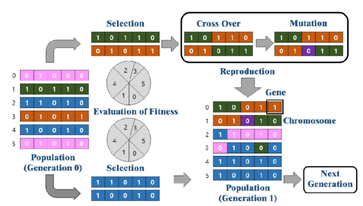
but how can we find this without knowing beforehand?

Evolutionary Algorithms

Evolutionary algorithms are optimization techniques, with genetic algorithms being a specific class. Genetic algorithms solve optimization problems using evolutionary principles and natural selection. They can help with Sudoku puzzles, machine learning, protein folding, and optimizing decision trees. Below is the process of a genetic algorithm [3].



In genetic algorithms (GA's), we begin with an initial population where each member is assigned a string of values, often represented as 0s and 1s. A fitness function measures how close an individual is to the target sum for our exponential sum expression. Higher fitness values indicate better solutions, with the fittest individuals more likely to reproduce and mate with other top individuals. Below is the process of a GA's (in 0s and 1s) [2].



We use crossover and mutation for reproduction, where an individual's DNA combines parts from both parents, and mutations switch 0s and 1s. We kill off those with lower fitness and keep breeding those that are most fit. We can incorporate birth, death, and mutation rates, which all have an effect on our population. After many generations of this process, it will converge to an optimal solution.

Field Symmetries/Automorphisms

A field, F, is a set where we have operations including $+$, $-$, $\times$, and $\div$. F has an identity, 1. All of these operations are mutually compatible. For exponential sums, we are living in a extension field of the rational numbers: $\mathbb{Q}[e^{2\pi i/N}]$. There is a specific symmetry for exponential sums that give us real numbers as our final sum value. Looking at an example of the a_k vector when $N = 7$, to get a real number sum value, our a_k has to be of the form:

$$[a_0, a_1, a_2, a_3, a_3, a_2, a_1].$$

We now look at the symmetries of field $\mathbb{Q}[e^{2\pi i/N}]$ in general. A field automorphism of a field F is a bijection

$$f : F \to F$$

such that

$$f(a \times b) = f(a) \times f(b),$$
$$f(a + b) = f(a) + f(b),$$

and

$$f(1) = 1.$$

Theorem 1. *The automorphisms of the field $F = \mathbb{Q}[e^{2\pi i/N}]$ are all of the form*

$$f\left(\sum_{k=0}^{N-1} a_k e^{2\pi i k/N}\right) = \sum_{k=0}^{N-1} a_{\sigma(k)} e^{2\pi i k/N}$$

where here σ is a permutation of $0, 1, 2, \ldots, N-1$ defined by $\sigma(k) = mk \mod N$ for some $m \in \mathbb{Z}$ relatively prime to N.

Incorporating Symmetries

At the end of our GA, we have two key values: the final target value, "z," and the converged sum value that our GA reached. Initially, our fitness function was $|\, z - \sum_{k=0}^{N-1} a_k e^{2\pi i k/N} \,|$ to measure the error bound. To simplify the range to be between 0 and 1, we redefined the fitness function as

$$e^{-|z-\sum_{k=0}^{N-1} a_k e^{2\pi i k/N}|^2}.$$

Imagine a landscape of peaks and valleys, with the highest peak being our target sum. There are other mountains that are the same height, but far away. Our function was giving us false peaks with identical fitness values but different converged results than the target sum. To fix this, we took into account the field's symmetries. Consider:

$$(\sum_{k=0}^{N-1} a_k e^{2\pi i k/N}) \cdot e^{2\pi i j/N} = \sum_{k=0}^{N-1} a_k e^{2\pi i (k+j)/N} = \sum_{l=0}^{N-1} a_{l-j} e^{2\pi i l/N}$$

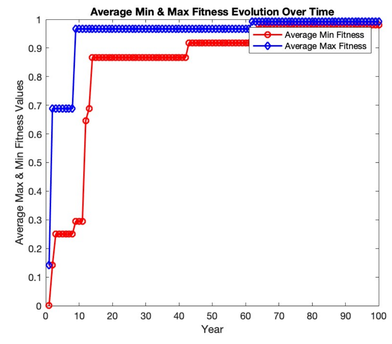
If you multiply an exponential sum by an exponential term, the result is another exponential sum, reflecting one of the symmetries of the original sum. By re-indexing the sum with an l, where $l = k + j$, the new sum is the original sum, but with its vector shifted to the left by l. Our final fitness function is

$$\max_j e^{-|z-\sum_{l=0}^{N-1} a_{l-j} e^{2\pi i l/N}|^2}.$$

This maximizes over the automorphisms applied to each member.

Results

Using the new fitness function, we can plot the average maximum and minimum fitness over 100 years, averaging across 50 experiments, 50 random target sums, and 50 random populations with an initial size of 5, and with $N = 201$.



Acknowledgements

I would like to thank the Math Department for funding this research project, & for their encouragement for students to pursue research and continue learning.

References

[1] David Angell. "Exponential Sums". In: *University of New South Wales Website* (Aug. 2024).

[2] Ki-Nam Hong Gun Park and Hyungchul Yoon. "Vision-Based Structural FE Model Updating Using Genetic Algorithm". In: *MDPI* (Feb. 2021), p. 6.

[3] Neelarghya. "Reinforcement Learning vs Genetic Algorithm — AI for Simulations". In: *Medium* (July 2021).