QUEST FOR 5D POLYTOPES USING A GENETIC ALGORITHM

Eldar Sultanow, Alexander Rahn, Malik Amir, and Andreas Hatziiliou

Abstract. Five-dimensional polytopes play a role in string theory, especially when they are reflexive. It is not trivial to find reflexive polytopes and there exists genetic approaches to finding them. We are implementing a genetic algorithm in Python and have found new five-dimensional polytopes.

1. Introduction

Before we start with the implementation and the actual polytop search, we want to explain a few basics.

1.1. **What is a polytope?** First of all, when we talk about polytopes, we mean lattice polytopes. In mathematics, a lattice in Euclidean space is a regular grid of points stretching infinitely in all directions. Imagine a 3D grid where points are spaced evenly in all three dimensions. These points have integer coordinates. In two dimensions, it would look like a regular grid of dots on a piece of graph paper.

A polytope is a general term that encompasses geometric shapes in any number of dimensions. In two dimensions, a polytope is a polygon (like a square or triangle), in three dimensions, it's a polyhedron (like a cube or pyramid), and in higher dimensions, we simply call them polytopes.

A lattice polytope is a polytope whose vertices are all located at the points of a lattice. That is, the vertices of a lattice polytope is integer-valued.

More formally, a d-dimensional lattice polytope P is the convex hull in $\mathbb{R}^d$ of a finite number of lattice points $x_1, \ldots, x_m \in \mathbb{Z}^d \subset \mathbb{R}^d$. We can list these points in the format of an $d \times m$ matrix $X = (x_1, \ldots, x_m)$ whose columns are the generators.

The first important quantity is the set of vertices $\{v_1, \cdots, v_{N_v}\}$ where the number of such vertex points denotes $N_v(P) \leq m$. It is less than or equal to the number of lattice points of the polytope as one could have, for example, three colinear lattice points the middle one not being included in the vertex set. We can combine these vertices into an $d \times N_v$ matrix $V = (v_1, \cdots, v_{N_v})$ called the vertex matrix.

2010 *Mathematics Subject Classification.* 52B05.
Key words and phrases. Polytopes, Calabi-Yau Manifold, Genetic Algorithm.

Now, define $H = \{x \in \mathbb{R}^d \mid u \cdot x = b\}$ be a hyperplane where $u \in \mathbb{Z}^d$ is a primitive inner normal and $b \in \mathbb{Q}$ is some rational number ("Hesse normal form" [1, p. 28,71], [2, p. 369]). "Primitive" means the coordinates of the normal are coprime [3] (comparable to the concept of an unit vector). "Inner" means that the normal is directed towards the inside of the polytope, meaning in the direction of the origin. We call a hypersurface valid if the polytope P is contained in the negative half-space (if $u \cdot x \leq b$ for all $x \in P$).

We call a face of P to be the intersection of P with all valid hyperplanes H. A facet is a face of dimension $d - 1$.

1.2. **What is the lattice distance?** The lattice distance between a lattice hyperplane $H \subset \mathbb{R}^d$ and a lattice point $x \in \mathbb{Z}^d$ is $\operatorname{dist}(x, H) \coloneqq |f(x)|$, where f is a primitive functional with $f(H) = 0$ [4, p. 50].

We can also say, the lattice distance $\operatorname{dist}(x, F)$ of a lattice point x of a polytope P from a facet F of P is defined as $\langle u, x \rangle - b = u \cdot x - b$, where $u \in \mathbb{Z}^d$ is a primitive inner normal of F and the hyperplane H spanned by F is given as $H = \{x \in \mathbb{R}^d \mid u{\cdot}x = b\}$ [5].

1.3. **What makes a lattice polytope reflexive?** The reflexive property of a polytope is given if the polytope has only one interior point and the lattice distance of each polytope's facet to this interior point is 1.

Hence we want polytopes with the following two properties

- The Interior Point (IP) property. A polytope satisfies the IP property if its only interior lattice point is the origin $O = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.
- The lattice distance between each facet of the polytope and its sole interior point, namely the origin, is 1.

Two polytopes P and Q with the same number of vertices are considered equivalent if their vertices are related by some integer linear transformation plus permutation matrix. An easy example to thing about is taking a square around the origin in $\mathbb{Z}^2$ and rotating it by 45 degrees. The resulting shape is a diamond around the origin but in essence its the same polytope as the original one.

Berglund, He, Heyes, Hirst, Jejjala, and Lukas [6] have demonstrated how a genetic approach can be used to identify new reflexive polytopes. The authors have found various polytopes and published their data set on GitHub [7]. We take up this idea and embark on a search for new five-dimensional polytopes.

For the quest of reflexive polytopes by genetic approach, we choose the following initial parameters :

- The dimension $d = 5$.
- The number of vertices of the polytope m. This will serve to organise points of P in a $5 \times m$ matrix.

- The value of x_{min} and ν which will serve as the range for the variables x_a^i of the matrix where $x_a^i \in \{x_{min}, \cdots, x_{min} + 2^\nu + 1\}$. This is simply a range for the size of the coordinates of the vertices of P where ν is the number of bits used for each matrix entry. This serves to define the environment E which will be the space of all $d \times m$ matrices with interval values in this range (phenotypes).

Next, compute the phenotype-genotype map (PGRM). Each integer x_a^i is converted into a bit string of length ν and concatenating these leads to a bit string of length $d_{bits} = dm\nu$ which describes the entire matrix of P. This map is a bijection and the environment containts a total of $2^{d_{bits}} = 2^{dm\nu}$ possible states.

Now, we want to define the probability distribution p_n and fitness function f. We use a flat distribution, that is every matrix/bit string in E has equal probability. For f we define the following fitness function where we start choosing the weights $\omega_1 = \omega_2 = 1$:

$$f(P) = \omega_1(IP(P) - 1) + \omega_2 \left(\sum_{F \in P} \text{dist}(O, F) - 1 \right)$$

Notice that $O = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ is the origin and the genetic algorithm will always have $f(P) \leq 0$ and will be equal to 0 exactly when the polytope P is reflexive. We set this to be our terminal state.

2. Implementing the genetic algorithm

To develop a solid algorithm with the possibility of adjustments on several ends, no Genetic Algorithm Library was used. Instead, the whole structure was built up from the scratch.

The current first version, does not implement specific mutation or selection methods which will be available in future versions. However, the fact of a first successfull finding of a reflexive polytope proofs the functionality of the underlying approach.

To be able to use a Genetic Algorithm for solving a Problem, a proper representation needs to be defined. For this, the choice fell on a matrix, which holds the information about the position of the vertices of the polytope.

A proper structure is provided in form of objects. The representation-matrix is part of the Chromosome object. A Generation consists of n Chromosomes. m Generations build the Population.

This structure is the base of the algorithm and builds up the overall solution in a top-down approach.

After defining basic information like the number of Generations, the number of Chormosomes per Generation, the dimension of the polytope and some more numbers, the

Population gets initialized. Case one: A first Generation is handed over while initializing the object, new Generations are generated through crossover of the existing polytopes.

Case two no first Generation is handed over, so n representation-matrixes of polytopes are randomly created and used as first Generation. This randomly generated Generation marks the start Generation of the Genetic Algorithm and is used to create new Generations through crossover. After the first Generation is set, all Chromosomes are sorted by their Fitness.

The fitness is based on the idea of Berglund et al. [6]. We penalize the fitness value if the polytope has more than one interior point and if the lattice distance of a vertex to the polytope's interior point is not equal to 1, see Listing 1.

```
def calc_fitness(vertices):
    ip_count = len(enumerate_integral_points(qhull(vertices)))
    distances = compute_distances(vertices)
    result = 0
    if ip_count > 1:
        result -= 1
    for d in distances:
        result -= abs(d-1)
    return result
```

Listing 1. Fitness function for searching reflexive polytopes

To create a new Generation through crossover, the previous Generation must be copied with deepcopy. After deleting the last half with worse fitness values than the first half, a random selection out of the remaining half is performed to get two parents, which create one new child through crossover. For the crossover a random one-point crossover function is used which generates a random split point and combines the two chromosomes in a new way. The new created solutions are added to the reduced Generation until it is filled up again.

This process is repeated, until the set number of Generations is created.

3. Preliminary results

Without any optimization of our genetic algorithm we already found new reflexive polytopes with 7 vertices, which is not included in the data set [7] as of 2024-01-01.

The matrix containing the vertices of one newly identified polytope is given by equation (1) and the vertices of another newly found polytope is given by equation (2).

$$P_1 = \begin{pmatrix} -2 & 1 & 4 & 1 & -3 & 1 & -2 \\ 3 & -2 & -4 & 2 & 3 & 2 & -1 \\ 1 & -1 & 1 & 3 & -1 & 3 & -3 \\ -3 & 2 & 2 & 0 & -2 & -3 & 0 \\ 0 & 0 & -2 & -1 & 2 & -1 & 2 \end{pmatrix} \tag{1}$$

$$P_2 = \begin{pmatrix} 4 & 1 & 4 & 1 & -3 & 1 & -2 \\ -2 & -2 & -4 & 2 & 3 & 2 & -4 \\ 3 & -1 & 1 & 3 & -1 & 3 & -5 \\ 2 & 2 & 2 & 0 & -2 & -3 & 2 \\ -3 & 0 & -2 & -1 & 2 & -1 & 1 \end{pmatrix} \tag{2}$$

The interior point of both polytopes is $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}$. Figure 1 displays these two newly found polytopes, projected from 5D into 3D.

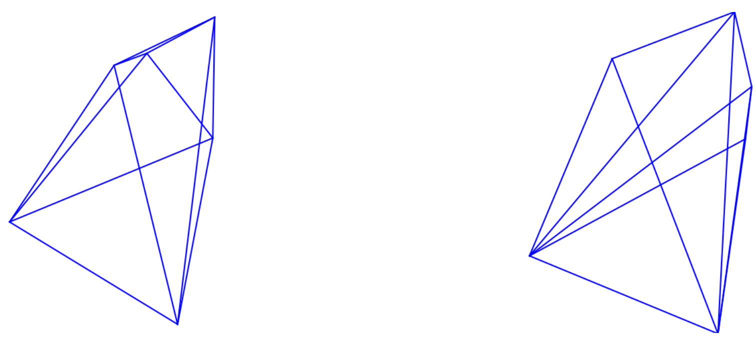

FIGURE 1. Newly found polytopes, the one given by matrix (1) on the left and the other one given by matrix (2) on the right

The reflexivity feature of this polytope can be verified using Sage as demonstrated by Listing 2 or using the CYTools framework [8] as shown by Listing 3, both given in the appendix A. The complete code is available at GitHub [9].

4. Conclusion and outlook

In the next steps, we optimize the genetic search algorithm and increase the search space to larger polytopes, that is, we expand the search for polytopes that have larger coordinates.

Another next step is evolving the visualization framework to provide a better and unambiguous graphical representation of the results. Barnette [10] as well as Wang, Yu, Chung, Gdawiec, and Ouyang [11], for instance, provide promising approaches that we could build on.

Appendix A. Listings

```
p = LatticePolytope([(-2, 3, 1, -3, 0), (1, -2, -1, 2, 0), (4, -4, 1,
    2, -2), (1, 2, 3, 0, -1), (-3, 3, -1, -2, 2), (1, 2, 3, -3, -1),
    (-2, -1, -3, 0, 2)])
p.is_reflexive()
```

Listing 2. Verify the polytope's reflexivity using Sage

```
from cytools import Polytope
vertices = [[-2, 3, 1, -3, 0],
 [1, -2, -1, 2, 0],
 [4, -4, 1, 2, -2],
 [1, 2, 3, 0, -1],
 [-3, 3, -1, -2, 2],
 [1, 2, 3, -3, -1],
 [-2, -1, -3, 0, 2]]
p = Polytope(vertices)
print(p.is_reflexive())
```

Listing 3. Verify the polytope's reflexivity using CYTools

References

[1] Klemens Burg, Herbert Haf Friedrich Wille, and Andreas Meister. *Höhere Mathematik für Ingenieure*, volume 2. Springer, 7 edition, 2012.

[2] Günter Bärwolff. *Höhere Mathematik für Naturwissenschaftler und Ingenieure*. Springer, 3 edition, 2017.

[3] Winfried Bruns, Joseph Gubeladze, and Mateusz Michałek. Quantum jumps of normal polytopes. *Discrete & Computational Geometry*, 56:181–215, 2016.

[4] Mónica Blanco. Lattice distances in 3-dimensional quantum jumps. In Alexander M. Kasprzyk and Benjamin Nill, editors, *Interactions with Lattice Polytopes*. Springer, 2017.

[5] Christian Haase, Benjamin Nill, and Andreas Paffenholz. Lecture notes on lattice polytopes. https://www2.mathematik.tu-darmstadt.de/~paffenholz/daten/preprints/20210628_Lattice_Polytopes.pdf, 6 2021. draft of June 28, 2021.

[6] Per Berglund, Yang-Hui He, Elli Heyes, Edward Hirst, Vishnu Jejjala, and Andre Lukas. New calabi-yau manifolds from genetic algorithms. *arXiv preprint arXiv:2306.06159*, 2023.

[7] Elli Heyes. Polytope-generation. https://github.com/elliheyes/Polytope-Generation/tree/main/Data, 2023. Accessed: 2023-12-29.

[8] Mehmet Demirtas, Andres Rios-Tascon, and Liam McAllister. Cytools: A software package for analyzing calabi-yau manifolds. *arXiv preprint arXiv:2211.03823*, 2022.

[9] Eldar Sultanow. Polytopes. https://github.com/Sultanow/polytopes, 2023. Accessed: 2023-12-29.

[10] David Barnette. A completely unambiguous 5-polyhedral graph. *Journal of Combinatorial Theory*, 9:44–53, 1970.

[11] Xingchang Wang, Tao Yu, Kwokwai Chung, Krzysztof Gdawiec, and Peichang Ouyang. Stereographic visualization of 5-dimensional regular polytopes. *Symmetry*, 11:44–53, 2019.

Eldar Sultanow, Capgemini, Bahnhofstrasse 30, 90402 Nuremberg, Germany
Email address: eldar.sultanow@capgemini.com

Alexander Rahn
Email address: alrahn@gmx.de

Malik Amir, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
Email address: malik.amir@epfl.ch

Andreas Hatziiliou, University of British Columbia, Department of Mathematics, 1984 Mathematics Rd, Vancouver BC, Canada
Email address: ahatzi@math.ubc.ca