3rd *BYMAT Conference* (2020)

Towards a database of isogeny graphs

Enric Florit Zacarías
Universitat de Barcelona
efz1005@gmail.com

Gerard Finol Peñalver
Universitat Rovira i Virgili
gerardfinol@gmail.com

Abstract: In number theory, it is often productive to gather arithmetic data in order to conjecture new results and discover unknown behaviour. The most notable modern case of this is the LMFDB, which contains lots of information on arithmetically interesting objects such as fields, algebraic curves and modular functions. Despite such a large collection of data, the isogeny-based cryptography community still lacks a range of examples of supersingular isogeny graphs. This work is a first attempt at generating these examples for genus 1, and it involves exploring elliptic curve isogenies and computing some of their graph invariants.

Resumen: En teoría de números, suele ser productivo recabar datos aritméticos para poder conjeturar nuevos resultados y descubrir comportamientos desconocidos. El caso moderno más notable es el de la base de datos LMFDB, que contiene información sobre objetos de interés aritmético tales como cuerpos, curvas algebraicas o funciones modulares. A pesar de existir tal colección de datos, la comunidad de criptografía basada en isogenias todavía carece de un repositorio de ejemplos de grafos de isogenias supersingulares. Este trabajo es un primer intento de generar estos ejemplos para género 1, e involucra explorar isogenias de curvas elípticas y computar algunos invariantes de dichos grafos.

Keywords: elliptic curves, isogeny graphs, distributed computing.

MSC2010: 11-04, 14K02.

Reference: Florit Zacarías, Enric, and Finol Peñalver, Gerard. "Towards a database of isogeny graphs". In: *TEMat monográficos*, 2 (2021): *Proceedings of the 3rd BYMAT Conference*, pp. 159-162. ISSN: 2660-6003. URL: https://temat.es/monograficos/article/view/vol2-p159.

1. Introduction

This paper is the starting point of a project to systematically produce data on isogeny graphs. Supersingular isogeny graphs of elliptic curves are used in proposed postquantum protocols, and having examples of them can help to further experiment with them. Our end goal is to investigate higher-dimensional abelian varieties over finite fields, and this is the first step to produce a framework for the task.

Not only we produce adjacency matrices of isogeny graphs, but we also want to list their graph invariants. Some quantitative work has already been done in [1], and we follow them to compute several of our metrics. We have used SageMath 9 over Python 3 [6] for our purposes.

2. Elliptic curves and isogeny graphs

An *elliptic curve* over a finite field $\mathbb{F}_q$ of characteristic $p \neq 2, 3$ is given by an equation

$$E\colon\ y^2 = x^3 + Ax + B,\ A, B \in \mathbb{F}_q$$

satisfying $4A^3 + 27B^2 \neq 0$. Such a curve has a group structure, displaying the simplest examples of abelian varieties. An *isogeny* between two elliptic curves is an algebraic map $E \to E'$ which is compatible with the group structures. Isogenies are characterised by the properties of being surjective and having finite kernel. The *degree* of a separable isogeny is the size of its kernel. If $\deg(\phi) = \ell$, we say ϕ is an ℓ-isogeny. An isomorphism is the case of an isogeny with trivial kernel.

Two elliptic curves are isomorphic over $\bar{\mathbb{F}}_q$ if and only if they have the same *j-invariant*, defined as

$$j(E) = 1728\frac{4A^3}{4A^3 + 27B^2}.$$

For each $j \in \bar{\mathbb{F}}_q$, there is an elliptic curve with that invariant, which we denote by E_j.

An elliptic curve is said to be *supersingular* if it has no p-torsion points. Hence, the *supersingular isogeny graph* $\Gamma_1(\ell; p)$, with $\ell \neq p$ two different primes, is defined as follows:

(i) Its vertices are the j-invariants of supersingular elliptic curves over $\bar{\mathbb{F}}_p$. These j-invariants are all in $\mathbb{F}_{p^2}$, and so they can be represented by two integers modulo p.

(ii) Given two vertices j and j', ϕ is an edge from j to j' if there is an ℓ-isogeny $\phi\colon E_j \to E_{j'}$. Multiple edges are allowed, although they are fairly rare in the genus 1 case.

For each ℓ-isogeny $\phi\colon E_j \to E_{j'}$ there is always a dual ℓ-isogeny $\hat{\phi}\colon E_{j'} \to E_j$, so we can regard $\Gamma_1(\ell; p)$ as being an undirected graph.

We can find a supersingular j-invariant in $\mathbb{F}_{p^2}$ in $\tilde{O}((\log p)^3)$ using Bröker's algorithm [3]. The graph $\Gamma_1(\ell; p)$ is always connected, so we can easily list all of its vertices with an exploration algorithm.

There are at least two known methods to compute an isogeny [2]. However, to compute the number of edges in $\Gamma_1(\ell; p)$ from j to j' it is sufficient to factor a modular polynomial. This allows us to work without equations for the curves E_j and $E_{j'}$, which would potentially require working over a larger finite field. Fix a prime p, and let N be any non-zero integer coprime with p. The Nth *modular polynomial* $\Phi_N(X, Y)$ is the equation that defines the planar model of the modular curve $X_0(N)$ classifying elliptic curves over $\mathbb{C}$ with a cyclic group of order N. The function field of this curve is $\mathbb{C}(j(\tau), j(N\tau))$, and so two curves E_j and $E_{j'}$ over $\mathbb{C}$ have an N-isogeny between them whenever $\Phi_N(j, j') = 0$. In fact, one can prove that $\Phi_N(X, Y) \in \mathbb{Z}[X, Y]$. Reducing the polynomial modulo p, we get the main result for our purposes: E and E' over $\bar{\mathbb{F}}_q$ are N-isogenous via a cyclic isogenies if, and only if, $\Phi_N(j(E), j(E')) = 0$.

Therefore, given $j = j(E)$, the N-neighbors of E are given by the roots of $\Phi_N(j, Y) \in \mathbb{F}_{p^2}[Y]$. In the case $N = \ell$ prime, this polynomial has at most $\ell + 1$ distinct roots (in general, the number of roots is given by Dedekind's ψ function).

2.1. Graph properties

Once we have the list of nodes of $\Gamma_1(\ell; p)$ and its adjacency matrix, we want to compute several of its properties, which we now explain.

(i) *Diameter and largest eigenvalues.* The graph $\Gamma_1(\ell; p)$ is $(\ell+1)$-regular and almost-undirected (i.e., it is undirected at every vertex except for a bounded number of them). Therefore, its diameter can be controlled by the eigenvalues of the adjacency matrix. More precisely, we know that the eigenvalues can be ordered as $\ell + 1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n > -(\ell+1)$.

If we fix the prime ℓ and let $\lambda_\star(p) = \max\{|\lambda_2|, |\lambda_n|\}$, the diameter of the family $\{\Gamma_1(\ell; p)\}_p$ grows like $O(\log p)$ as long as there exists some fixed constant $\Lambda_\ell < \ell + 1$ with $\lambda_\star(p) \leq \Lambda_\ell$ for all p. This is indeed the case for supersingular graphs of elliptic curves (they have the Ramanujan property), but the result for higher-dimensional varieties is still conjectural [5].

(ii) *Size of the spine.* The spine of $\Gamma_1(\ell; p)$ is the induced subgraph of vertices that are defined over $\mathbb{F}_p$. Knowing the structure of the spine is useful since it tends to be a very small subgraph where finding paths is simpler. If we are able to solve that particular problem, then finding a path between any two vertices reduces to finding paths to the spine.

(iii) *Number of isogenous conjugate pairs.* Each j-invariant outside of the spine, $j \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$, has a Frobenius conjugate j^p. This corresponds to the Frobenius (inseparable) isogeny $E \to E^{(p)}$, given by $(x, y) \mapsto (x^p, y^p)$. An ℓ-isogenous conjugate pair is a pair of vertices (j, j^p) connected by an ℓ-degree isogeny.

3. Distributed computation

Due to the great computational cost of calculating the data, the task has to be performed in parallel. Parallel programming on a single computer has been useful for graphs with relatively small p, where they could be calculated using a typical 4-core personal computer in a reasonable time. In order to calculate the graphs with $p \approx 30\,000$, we needed to scale the computational power beyond a single computer and use distributed computing. Therefore, the calculations have been carried out in a parallel and distributed way, having more than 500 threads of parallel execution in a distributed way[1].

3.1. Graph computation procedure

The computation of each graph and its properties has been divided in five sequential stages (i.e., a stage can only be run after the previous ones). We now describe them.

The first step to start working with graphs is *computing the nodes* of the ℓ-isogeny graph, so for a given p we want to compute a list with all the nodes from $\Gamma_1(\ell; p)$. To discover the nodes we use a slight modification of the breadth-first search algorithm (BFS) starting from an initial node. Given a node j, we obtain all its neighbours by factoring $\Phi_\ell(j, Y)$. Because the node list depends exclusively on p, and for efficiency reasons, we explore the graph $\Gamma_1(\ell; p)$ with $\ell = 2$. The BFS algorithm is not well suited to be run in parallel, so parallelization has been achieved by just exploring multiple graphs simultaneously.

Checks over the node list. On the one hand, we know we must have $\left[\frac{p-1}{12}\right] + \varepsilon$ nodes in $\Gamma_1(\ell; p)$ (with $\varepsilon = 0, 1, 1, 2$ according to $p \equiv 1, 5, 7, 11 \mod 12$). On the other hand, we can test any given node for supersingularity with SageMath's function `E.is_supersingular`. Using these two facts, we can guarantee that the computed node list is complete and correct.

Computation of the adjacency matrix of $\Gamma_1(\ell; p)$. Given p and ℓ, we compute all the neighbours of each node in $\Gamma_1(\ell; p)$. This task is highly parallelizable, since it is enough to split the list of nodes into batches of similar size and assign one batch to every thread of execution. Once we have the list of neighbours for every node it can be easily converted to the adjacency matrix.

[1] Code can be found at https://github.com/gfinol/IsogenyGraph.

Checks on adjacency matrix. We check that the matrix is square, has correct dimensions and all nodes have out-degree $\ell + 1$. It is important to notice that these are sanity checks to discard possible errors on the computation rather than checks to prove the correctness of the whole matrix.

Finally, we *compute the graph metrics* using the adjacency matrices. Similarly to our other tasks, we compute them for several graphs simultaneously.

3.2. Lithops

To scale computational power beyond one machine we have used the Lithops[2] framework, which provides an API mimicking the Python multiprocessing library and allows us to execute our code transparently [7] in a distributed serverless environment without having a physical computer cluster nor having to manage one. Thanks to the similar APIs, the code can be executed in parallel on a single machine or distributed using FaaS by just changing the module import from multiprocessing to Lithops. This also allows us to use SageMath in a distributed environment.

4. Results and future work

We have computed all graphs $\Gamma_1(\ell; p)$ for primes $13 \leq p < 30\,000$ and degrees $\ell \in \{2, 3, 5, 7, 11\}$, along with the graph properties specified in Section 2.1. The data has been uploaded to Zenodo [4].

We have built a framework to compute examples for larger p and ℓ in the future, and that will also allow us to explore isogeny graphs of higher-dimensional abelian varieties. This will provide us with data to further confirm existing conjectures [5] on such graphs.

References

[1] Arpin, Sarah; Camacho-Navarro, Catalina; Lauter, Kristin; Lim, Joelle; Nelson, Kristina; Scholl, Travis, and Sotáková, Jana. "Adventures in Supersingularland". In: *arXiv e-prints* (2019). arXiv: 1909.07779 [math.NT].

[2] Bernstein, Daniel J.; De Feo, Luca; Leroux, Antonin, and Smith, Benjamin. "Faster computation of isogenies of large prime degree". In: HAL CCSD, 2020. https://doi.org/10.2140/obs.2020.4.39.

[3] Broker, Reinier. "Constructing supersingular elliptic curves". In: *Frontiers of Combinatorics and Number Theory* (2009).

[4] Florit, Enric and Finol, Gerard. *Isogeny graphs of supersingular elliptic curves.* Version 1.0.1. Zenodo, Nov. 2020. https://doi.org/10.5281/zenodo.4304044.

[5] Florit, Enric and Smith, Benjamin. "Automorphisms and isogeny graphs of abelian varieties, with applications to the superspecial Richelot isogeny graph". In: *arXiv e-prints* (2021). arXiv: 2101.00919 [cs.CR].

[6] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.1)*. https://www.sagemath.org. 2021.

[7] Sampe, J.; Garcia-Lopez, P.; Sanchez-Artigas, M.; Vernik, G.; Roca-Llaberia, P., and Arjona, A. "Toward Multicloud Access Transparency in Serverless Computing". In: *IEEE Software* 38.1 (2021), pp. 68–74. issn: 1937-4194.

[2] https://github.com/lithops-cloud/lithops