

Refutation of Spectral Graph Theory Conjectures with Search Algorithms

Milo Roucairol, Tristan Cazenave

LAMSADE, Université Paris Dauphine - PSL,
Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

Abstract

We are interested in the automatic refutation of spectral graph theory conjectures. Most existing works address this problem either with the exhaustive generation of graphs with a limited size or with deep reinforcement learning. Exhaustive generation is limited by the size of the generated graphs, and deep reinforcement learning takes hours or days to refute a conjecture. We propose to use search algorithms to address these shortcomings to find potentially large counter-examples to spectral graph theory conjectures in seconds. We apply a wide range of search algorithms to a selection of conjectures from Graffiti. Out of 13 already refuted conjectures from Graffiti, our algorithms are able to refute 12 in seconds. We also refute conjecture 197 from Graffiti, which was open until now.

Keywords

Monte Carlo Search, Spectral Graph Theory, Conjecture, Refutation

1. Introduction

Monte Carlo search algorithms have proven to be powerful as game-playing agents, with recent successes like AlphaGo [1]. These algorithms have the advantage of only needing an evaluation function for the final state of the space they explore.

Graph conjectures are propositions on graph classes (any graph, trees, K_n – free...) that are suspected to be true and are awaiting proof or a refutation. They lend themselves well to computer-assisted proofs, as finding a counterexample can be tedious to do manually. Spectral graph conjectures are appropriate for automated refutation because the property can often be directly turned into the evaluation function that can take many different values. Thanks to software like Auto-GraphiX [2] and Graffiti [3], there are plenty of such conjectures.

Adam Zsolt Wagner showed that one could find explicit counterexamples using deep reinforcement learning of a policy with the deep cross-entropy method [4]. Wagner was able to refute three conjectures from Graffiti. Wagner’s method trained a neural network with reinforcement learning for each conjecture, learning to build graphs tailored to the refutation of each conjecture. The refutation of a conjecture with Wagner’s method takes hours or days. Using search algorithms, it was possible to refute conjectures 2.1 and 2.4 from the Wagner paper in one second and conjecture 2.3 in 291 seconds [5].

A usual way to find counterexamples to graph theory conjectures is to generate all the graphs smaller than a given size and to verify the conjecture for all these small graphs [6]. This exhaustive generation limits the size of the graphs that can be tested. On the contrary, our search algorithms can rapidly generate larger refutation graphs.

In this paper, the search algorithms will play the game of refuting conjectures by building counterexamples edge by edge. We build on our previous paper [5] by testing many more conjectures and by comparing more search algorithms.

IASE 2025

✉ milo.roucairol@dauphine.eu (M. Roucairol); tristan.cazenave@dauphine.fr (T. Cazenave)

🌐 <https://www.lamsade.dauphine.fr/~cazenave/index.php> (T. Cazenave)

🆔 0000-0002-7794-5614 (M. Roucairol); 0000-0003-4669-9374 (T. Cazenave)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

First, we will present the refutation of graph theory conjectures, then the different algorithms we use to explore the problem space, after that, the procedure we use to build graphs and the game rules, and finally, we expose our new results on multiple different conjectures.

2. Refutation of Graph Theory Conjectures

2.1. Graph Theory Conjectures

Refuting Graph theory conjectures can be a hard task to do manually. Imagining a large number of graphs and computing invariant or NP-hard problem values can be tedious and often results in a waste of time. Computers are designed to help with these score computations. The goal of automated conjecture refutation is to automate the exploration as well.

Graph theory conjecture can concern many different properties of graphs: existence, topology, flow, connectivity, cycle, minors, spectral... Here, we are interested in the spectral graph theory conjectures, necessitating matrix calculations only. The spectrum of a matrix is the set of its eigenvalues. Spectrum-related invariants on different types of graph-related matrices (adjacency, distance, Laplacian...) are the focus of spectral graph theory conjectures.

We propose to compare search algorithms for the refutation of various spectral graph theory conjectures. In [5], the conjectures from [4] were refuted hundreds of times faster, and the performances of the methods were compared. This paper experiments with a more diverse set of search algorithms, and we address all the remaining conjectures from Graffiti that do not involve solving an NP-hard problem. Graffiti is a well-known software that was used to generate a thousand spectral graph theory conjectures. It was the target of numerous works and discussions, sometimes involving some of the greatest graph theorists such as Paul Erdos, Laszlo Lovasz, Noga Alon, Noam Nisan. Conjectures from Graffiti were the subject of entire articles, with graph theorists dedicating weeks or months to them. Our program aims at refuting some of them in a much more efficient manner.

We decided to focus on this subset of Graffiti conjectures to avoid dependence on other algorithms and to reduce the number of conjectures.

2.2. Algorithms Used to Refute Graph Conjectures

The program by Wagner trained a deep neural network with cross-entropy to output a policy. The network is used to learn a policy from a state. The policy is used to generate a batch of graphs, it then evaluates the best graphs from the batch and trains the neural network using the graphs' scores. Batches are successively evaluated until the refutation of the conjecture.

Software used for automated conjecture generation, like Graffiti or Auto-GraphiX, also has its ways of automatically refuting conjectures to verify the easily refutable ones. Graffiti and its conjectures are detailed in Aouchiche and Hansen's survey [6]. It generates many conjectures in the form of inequalities between invariants. It then tests these conjectures on a database of graphs and discards the falsified ones. Then it checks if the inequalities are not implied by already known theorems and conjectures. If a conjecture passes these tests, it is proposed to scientists. "Written on the wall" collects almost 1000 conjectures from Graffiti, along with the discussions of many scientists. Graffiti's conjectures connect various topics in spectral graph theory, such as eigenvalues, adjacency, distance, Laplacian, and gravity matrices of graphs. These conjectures are the subjects of dozens of articles.

3. Materials and Methods

We decided to expand the experiments from [5] by using the same algorithms and adding two widely used Monte Carlo Search algorithms, another recent one, and another baseline simple algorithm on

more conjectures produced by Graffiti [6]. The search model used by the algorithms to build the graphs is the same as in [5]: starting from a single node, new nodes and edges are added to the graph until it reaches a target size. These additions of nodes and edges are the edges of the search tree.

- NMCS [7] uses nested levels of search with random search at the base level. It is already used in [5].
- NRPA [8] learns a playout policy with nested levels of best sequences. At the lowest level, it makes playouts with the learned policy. It is already used in [5].
- GBFS is a simple greedy algorithm opening the best state from a list, evaluating the children of this state, and inserting these children in the list according to their evaluation. It is already used in [5].
- BEAM search, another baseline greedy search heuristic, keeps the *width* best states after each step of expanding and going down the search tree.
- Upper Confidence bounds applied to Trees (UCT) as described by Kocsis and Szepesvári [9], which is the most widely used MCTS algorithm, and close to PUCT used in outstanding works such as Deepmind’s Alphazero [10].
- Rapid action value estimation (RAVE), a MCTS algorithm inspired by UCT and proposed by Silver and Gelly [11], then generalized as GRAVE by Cazenave [12] using a threshold on the number of playouts.
- Lazy Nested Monte Carlo Search (LNMCS), a variant of NMCS proposed by Roucairol and Cazenave [13] addresses a shortcoming of the NMCS. Before launching a costly lower-level LNMCS on a child state, it is evaluated with playouts and pruned if the evaluation is not satisfying enough, this way it avoids sinking the time budget in fruitless subtrees and allows the LNMCS to use higher starting levels than the NMCS.

These conjectures (in table 1) were selected because they did not require NP-hard problems to be solved to compute the score, thus lending themselves well to Monte Carlo Search and other optimization methods. All of the fitting conjectures from Aouchiche and Hansen’s survey [6] are included in our experiments. It is possible to tackle conjectures needing an NP-hard problem to be solved (like coloration) up to sizes 25-30, we might come back to such conjectures in future works.

4. Results

The experiments were made with Rust 1.59, on an Intel Core i5-6600K 3.50GHz, using a single core (but parallel processing is very accessible).

Our positive results in table 1 were obtained by launching each algorithm on each conjecture with a time budget of 15 minutes. In the Graffiti column, R means the conjecture was already refuted, and O means the conjecture is still open.

- The NMCS used a level of 3.
- The LNMCS used a level of 4, 3 playouts per evaluation, and a ratio of 0.8.
- The UCT used a constant of 1.
- The GRAVE and RAVE used a reference of 5.
- The BEAM search used a width of 10.

5. Discussion

As shown in table 1, we were unable to refute conjectures that were open except one (either 197 or 322). This means either that our algorithms are not strong enough, or that these conjectures are true. However, with the exception of Graffiti 140, most of the already refuted conjectures we tried were refuted by at least one algorithm among GRAVE, RAVE, NMCS, LNMCS, NRPA, GBFS, and UCT. Graffiti

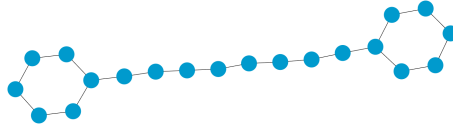


Figure 1: A counter-example of Graffiti 289 of size 20 (second largest eigenvalue \leq mean of the mean of all adjacent vertex degree for all nodes)

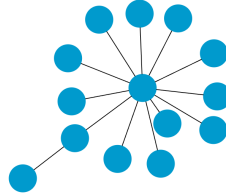


Figure 2: A counter-example of Graffiti 301 of size 14 (scope of positive eigenvalues \leq harmonic)

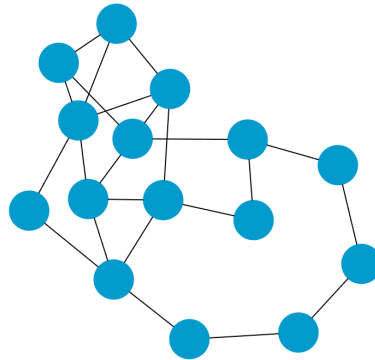


Figure 3: A counter-example of Graffiti 30 of size 15 (# positive distance eigenvalues \leq sum of temperatures)

140 refutation seems to be lost, 140 is said to be refuted by another work by Favaron, which we are unable to find.

Considering nonterminal states for evaluation helped immensely with the harder Graffiti 137, 139, 189, 289, and 301. Graffiti 289's repeating pattern may be an explanation as to why the NRPA succeeded where most other algorithms failed. The times, when displayed in the hundreds of seconds, are approximations, only GBFS and BEAM are deterministic, so the time is exact. In addition, NMCS, UCT, GRAVE, and RAVE were very unreliable at finding a counter-example for Graffiti 29, about once every 3 runs, when LNMCS was able to find a counter-example in less than 60 seconds in 9 out of 10 runs.

For Graffiti 189 and 195, we first searched in the "any graph" search space (instead of the type of graphs the conjecture is about), and then verified if the graph in question complied with the conjecture's definition.

We think the relative failure of GBFS on Graffiti 30 is partly due to the score function leading the algorithm to open nodes to smaller graphs first, while the MCTS algorithms try larger graphs immediately using playouts. Beam search requires a width of 80 or more to find a counterexample to this conjecture, so trying larger graphs is not sufficient on its own.

The beam search is very dependent on its width parameter, too high and it will not be able to explore deep in the search tree and find large counterexamples (on Graffiti 137), and too low, and it will not consider inferior intermediate states that would, in fact, lead to a counterexample (on Graffiti 29 and 30). That makes it an inferior choice for spectral graph conjecture refutation compared to GBFS.

In [5] Graffiti 137 was one of the 4 main conjectures refuted. Unfortunately, an erroneous definition of the Harmonic was used (see subsection 5.2). Graffiti 137 is more interesting than we initially thought. The smallest known counter-example featured 101 nodes and required a very specific structure; it was first refuted by Favaron et al. [14] using a mathematical proof and not directly a counter-example. Here

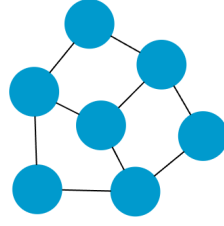


Figure 4: A counter-example of Graffiti 29 of size 7 (randic index \leq # negative eigenvalues)

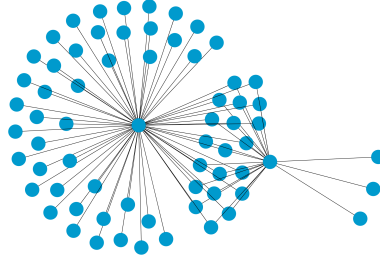


Figure 5: A counter-example of Graffiti 137 of size 67 (second largest eigenvalue \leq harmonic)

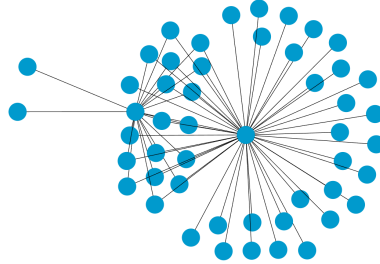


Figure 6: A counter-example of Graffiti 139 of size 50 (- second smallest eigenvalue \leq harmonic)

we managed to find a counter-example of size 67 (see figure 5) which does not follow the structure described, but closely resembles it, by descending the search space (opening graphs of increasing size, much like conjecture 2) almost directly with GBFS.

As mathematical proofs are complex, if the goal was to refute this one conjecture, then our algorithm was most likely faster. But another interesting result is that the class of graphs needed to refute Graffiti 137 by Favaron et al. was also used to refute two other conjectures (and could have been used to refute Graffiti 139, see figure 6). The search algorithms can provide insights to the mathematicians (they can provide insights even if the graph produced does not refute a conjecture as in Wagner's work [4] with conjecture 2.3). The other algorithms did not manage to find a counterexample in a reasonable time.

5.1. Graffiti 197 is refuted

In "Written on the wall", Graffiti 197 is defined as:

Graffiti 197. - 2nd smallest eigenvalue \leq range of eigenvalues of the gravity matrix.

Using the usual definition of the range, and not Aouchiche and Hansen's (see subsection 5.2), our algorithms find a counterexample with the cycle of length 17, see figure 7. Using Aouchiche and Hansen's definition of the range, we could not find any counterexample, but if this initial definition were to be used, then Graffiti 322 would be (too) easily refuted.

The second smallest eigenvalue is $\lambda_{n-1} \approx -1.9659$, and the range of eigenvalues of the gravity matrix is approximately 1.7035.

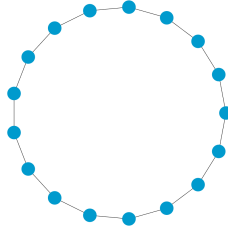


Figure 7: A counter-example of Graffiti 197 of size 17

Cycles of size 21 and 25 also refute Graffiti 197. We conjecture that any cycle of size $1 + 4 * n$ greater than or equal to 17 refutes this conjecture.

5.2. Erratum

We provide a copy of "Written on the Wall" on this project's github, which is otherwise hard to find:

- Graffiti 290 (open) was solved instantly using the definition of gravity in Aouchiche and Hansen's survey [6] (which prompted us to verify the definition). But it was seemingly impossible using the definition from Brewster et al. [15].

Since the definition of the gravity of a graph is not widespread, here is the correct definition of the gravity matrix from "Written on the Wall" (page 52) and Brewster et al. [15]:

Gravity matrix: The matrix (indexed by vertices of the graph) whose $(u, v)th$ entry is 0 if $u = v$ or if there is no path joining u to v ; otherwise is:

$$Gr(G) = (1/(n - 1))(d(u) * d(v))/d(u, v) \quad (1)$$

Where $d(u)$ is the degree of vertex u and $d(u, v)$ is the distance from vertex u to vertex v in the graph G of size n .

- In the same manner, with Aouchiche and Hansen's definition of the harmonic, Graffiti 140 was refuted by the graph with one edge connecting two vertices. The correct definition is featured in "Written on the Wall" (page 28) and in Favaron et al. [14]:

Harmonic: The harmonic of a graph is defined as:

$$Hc(G) = \sum_{uv \in E} \frac{2}{d(u) + d(v)} \quad (2)$$

Where E is the set of edges of a graph G .

- Using Aouchiche and Hansen survey's [6] definition of range, it appeared that our algorithms managed to refute Graffiti 322 (open) with a cycle graph of size 4. Here is the definition of Graffiti 322 from "written on the wall":

Graffiti 322. If G is a triangle-free graph then the Inverse Even \leq range of eigenvalues of Distance.

Inverse Even is defined as the sum over the vertices of the inverse of the number of vertices at an even distance from that vertex, ie:

$\sum_{v \in V} 1/Ev(v)$ where $Ev(v)$ is the number of vertices at even distance from vertex v .

Thus, the cycle of size 4 has 3 distinct distance eigenvalues and an Inverse Even of 4. An error with the definitions seems more likely than this conjecture being left open after dozens of articles with such a simple counterexample.

The range is defined as the number of distinct values in a vector in Aouchiche and Hansen’s survey and (likely inherited from) Favaron et al.’s work, with no further sources. The range of a vector is usually the difference between the minimum and maximum values. Thus, we think this definition is erroneous but we were not able to find the actual definition of the range used in "written on the wall". The results featured in table 1 for Graffiti 322 use the usual definition of range; either way, Graffiti 197 is refuted or Graffiti 322 is refuted.

- Graffiti 294 from Aouchiche and Hansen’s survey is Graffiti 295 from "written on the wall", we go by this identifier in table 1.
- Graffiti 140 is said to be refuted by Favaron et al. in "residue of a graph" [16] according to "written on the wall", but there is no mention of "harmonic", "deviation", or Graffiti 140 in that paper.

The general lack of sourcing, the loss of multiple articles, and the misleading or erroneous definitions are detrimental to future work on Graffiti conjectures. For instance, we suspect that Graffiti conjectures 28 and 209 may not be refuted as sources are nonexistent (28) or seemingly lost (209). We decided to include Graffiti 140 in the table, unlike 28 and 209, because we could find its refutation source, but we could not find its appearance in the source. Auto-GraphiX conjectures may be more favorable to continuing this work.

6. Conclusion

Search algorithms proved to be powerful ways of refuting conjectures from spectral graph theory, much faster than Wagner’s deep cross-entropy method [4]. We can identify three algorithms that seem more effective on the conjectures tested: GBFS, which is overall faster but was not able to refute conjectures in [5] and struggled on Graffiti 30. NRPA, which is very efficient when a repeating pattern refutes a conjecture. And LNMCS/NMCS, which managed to be slightly faster than NRPA on Graffiti 29 and 301, while being faster than the other MCTS algorithms. LNMCS did not outperform NMCS due to the nature of the search space: there rarely is a move that dooms all its child states. It seems that not one single algorithm is able to solve all of the sample of conjectures we selected, and using different approaches is required to adapt to the variety of problems in spectral graph theory conjecture refutation. We recommend trying GBFS, the NMCS, and then NRPA.

Trying to build trees even when the conjecture is applied to any graph can also be helpful, as it reduces the amount of possible builds greatly and focuses on a sub-type of graphs that often features extreme properties over the spectral invariants. It is inexpensive and should be tried first.

However, these methods present limits. Computing score functions that require eigenvalues on big trees (over 300 nodes) can be very costly. They are also dependent on the shape of the score function: a noisy score function with many local minima can be challenging, as well as a score function with more discrete results can lead to an absence of differentiation in the paths to explore. Conjectures requiring computing an NP-hard problem can also severely increase the computing time even for small graphs (30 vertices).

Despite these limitations, and depending on the definition of the gravity matrix, either our program was able to refute up to the state of the art (manually from graph theorists) the conjectures from Graffiti, or it refuted Graffiti 197, a previously open conjecture.

You can access the implementations of the conjecture and the Rust code used to refute them here:

<https://github.com/RoucairolMilo/refutationGBR>

Statements and Declarations

Thanks to Édouard Bonnet for fruitful discussions. Thanks to Craig E. Larson and Ermelinda DeLaViña for answering some of our questions and providing "written on the wall" to us.

Funding

This work was supported in part by the French government under the management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute).

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

References

- [1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, *Nature* 529 (2016) 484–489.
- [2] P. Hansen, G. Caporossi, AutoGraphiX: An Automated System for Finding Conjectures in Graph Theory, *Electronic Notes in Discrete Mathematics* 5 (2000) 158–161. URL: <https://www.sciencedirect.com/science/article/pii/S1571065305801519>. doi:10.1016/S1571-0653(05)80151-9.
- [3] E. Delavina, Some History of the Development of Graffiti, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science 69: Graphs and Discovery, 2005, pp. 81–118.
- [4] A. Z. Wagner, Constructions in combinatorics via neural networks, arXiv:2104.14516 [cs, math] (2021). URL: <http://arxiv.org/abs/2104.14516>, arXiv: 2104.14516.
- [5] M. Roucairol, T. Cazenave, Refutation of spectral graph theory conjectures with monte carlo search, in: International Computing and Combinatorics Conference, Springer, 2022, pp. 162–176.
- [6] M. Aouchiche, P. Hansen, A survey of automated conjectures in spectral graph theory, *Linear Algebra and its Applications* 432 (2010) 2293–2322. URL: <https://www.sciencedirect.com/science/article/pii/S0024379509003061>. doi:10.1016/j.laa.2009.06.015.
- [7] T. Cazenave, Nested Monte-Carlo Search, in: C. Boutilier (Ed.), IJCAI, 2009, pp. 456–461.
- [8] C. D. Rosin, Nested rollout policy adaptation for monte carlo tree search, in: In IJCAI, 2011, pp. 649–654.
- [9] L. Kocsis, C. Szepesvári, Bandit based Monte-Carlo planning, in: 17th European Conference on Machine Learning (ECML’06), volume 4212 of LNCS, Springer, 2006, pp. 282–293.
- [10] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, D. Hassabis, Mastering chess and shogi by self-play with a general reinforcement learning algorithm, *CoRR* abs/1712.01815 (2017). URL: <http://arxiv.org/abs/1712.01815>. arXiv:1712.01815.
- [11] S. Gelly, D. Silver, Monte-carlo tree search and rapid action value estimation in computer go, *Artificial Intelligence* 175 (2011) 1856–1875.
- [12] T. Cazenave, Generalized rapid action value estimation, in: 24th International conference on artificial intelligence, 2015, pp. 754–760.
- [13] M. Roucairol, T. Cazenave, Solving the hp model with nested monte carlo search, arXiv preprint arXiv:2301.09533 (2023).
- [14] O. Favaron, M. Mahéo, J.-F. Saclé, Some eigenvalue properties in graphs (conjectures of graffiti – ii), *Discrete Mathematics* 111 (1993) 197–220. URL: <https://www.sciencedirect.com/science/article/pii/0012365X9390156N>. doi:[https://doi.org/10.1016/0012-365X\(93\)90156-N](https://doi.org/10.1016/0012-365X(93)90156-N).
- [15] T. L. Brewster, M. J. Dinneen, V. Faber, A computational attack on the conjectures of graffiti: New counterexamples and proofs, *Discrete Mathematics* 147 (1995) 35–55.
- [16] O. Favaron, M. Mahéo, J.-F. Saclé, On the residue of a graph, *Journal of Graph Theory* 15 (1991) 39–64.

Table 1

Times in seconds to obtain a refutation by applying every algorithm on selected Graffiti conjectures.

Graffiti:

R means the conjecture was already refuted (as reported in [6]),

O means the conjecture is open to be proved or refuted.

Size:

If there is a "+", it is the size at which we start to find counterexamples with our algorithms. Otherwise, it is the size we built graphs up to.

Graph type:

It is the types of graphs we used in our searches. "any" means there was no restriction on the graph built, $K3 - free$ means there is no clique of size 3, "girth ≥ 5 " means the shortest cycle must be longer than 5, "tree" means the graph must be a tree. We tried both generating trees and then any graph on multiple conjectures.

A 0 in the time section means that the conjecture was refuted in less than a second.

| Graffiti | Size | Graph type | NMCS | LN MCS | NRPA | UCT | GBFS | BEAM | GRAVE | RAVE |
|----------|------|----------------|------|--------|------|-----|------|------|-------|------|
| 166 R | 5+ | any | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 321 R | 4+ | $K3 - free$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 189 R | 5+ | any | 0 | 0 | 0 | 2 | 10 | 4 | 2 | - |
| 289 R | 20+ | girth ≥ 5 | 600 | 600 | 200 | - | 6 | 102 | - | - |
| 302 R | 6+ | tree | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 715 R | 16+ | tree | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 29 R | 7+ | any | 2 | 2 | 10 | 5 | 0 | - | 2 | 1 |
| 30 R | 12+ | any | 0 | 0 | 0 | 0 | 311 | - | 0 | 0 |
| 301 R | 14+ | tree | 2 | 2 | 4 | - | 0 | 0 | 7 | 2 |
| 137 R | 67+ | any | - | - | - | - | 513 | - | - | - |
| 139 R | 50+ | any | - | - | - | - | 36 | - | - | - |
| 711 R | 5+ | any | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 |
| 197 O | 17+ | $K3 - free$ | 30 | 30 | 5 | - | 0 | 4 | - | - |
| 140 R | 50 | any & tree | - | - | - | - | - | - | - | - |
| 290 O | 50 | girth ≥ 5 | - | - | - | - | - | - | - | - |
| 284 O | 50 | girth ≥ 5 | - | - | - | - | - | - | - | - |
| 195 O | 50 | any | - | - | - | - | - | - | - | - |
| 21 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 39 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 143 O | 100 | any & tree | - | - | - | - | - | - | - | - |
| 154 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 20 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 40 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 254 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 262 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 712 O | 50 | any & tree | - | - | - | - | - | - | - | - |
| 198 O | 50 | any | - | - | - | - | - | - | - | - |
| 714 O | 100 | any | - | - | - | - | - | - | - | - |
| 219 O | 50 | $K3 - free$ | - | - | - | - | - | - | - | - |
| 322 O | 50 | $K3 - free$ | - | - | - | - | - | - | - | - |
| 292 O | 50 | girth ≥ 5 | - | - | - | - | - | - | - | - |
| 295 O | 50 | girth ≥ 5 | - | - | - | - | - | - | - | - |
| 129 O | 50 | any | - | - | - | - | - | - | - | - |
| 698 O | 50 | any | - | - | - | - | - | - | - | - |
| 252 O | 50 | any | - | - | - | - | - | - | - | - |