Discrete Applied Mathematics 201 (2016) 172–181

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Integer sequence discovery from small graphs

Travis Hoppe [a,*], Anna Petrone [b]

[a] National Institutes of Health, National Institute of Diabetes and Digestive and Kidney Diseases, 9000 Rockville Pike, Bethesda, MD, 20892, United States

[b] University of Maryland, Department of Civil Engineering, United States

ARTICLE INFO

Article history:
Received 16 September 2014
Received in revised form 6 July 2015
Accepted 19 July 2015
Available online 18 August 2015

Keywords:
Graph
Integer sequence
Database
Invariant
Generator

ABSTRACT

We have exhaustively enumerated all simple, connected graphs of a finite order and have computed a selection of invariants over this set. Integer sequences were constructed from these invariants and checked against the Online Encyclopedia of Integer Sequences (OEIS). 141 new sequences were added and six sequences were extended. From the graph database, we were able to programmatically suggest relationships among the invariants. It will be shown that we can readily visualize any sequence of graphs with a given criteria. The code has been released as an open-source framework for further analysis and the database was constructed to be extensible to invariants not considered in this work.

Published by Elsevier B.V.

1. Introduction

There is a long history of public graph databases. Databases originally found only in print, such as the *Atlas of Graphs* [31], have rapidly expanded to the electronic medium. These databases range from those of mathematical and algorithmic interest [11,5], to those cataloging structures found in the applied sciences such as ChemSpider [30], RNA topologies [14,21] or social databases [22,4,29]. Due to the rapid growth in the number of unique isomorphic graphs, the currently available databases are specialized in the number of graphs considered; a judicious choice often restricts the study to an interesting and more manageable subset.

We proceed with the assumption however, that *a priori* all graphs could be interesting given the right question. This is similar to the GraPHedron project [27], which attempts to formulate conjectures by searching for graphs bounded by an inequality or constraint. Our objective is more elementary. We aim to compute a large, comprehensive database of graphs and their respective invariants. Such a database will allow new forms of discovery, some of which will be directly explored in this paper. For example, the integer sequences formed by the invariants can be systematically explored and compared to those already known. These sequences, and the set of graphs that belong to them, can be used to explore a basic set of relations among the invariants. Since the input to GraPHedron consists of a set of invariants, a larger input set will amplify the predictive power. Additionally, the creation of a large, centralized database will serve as a useful reference for benchmarking various algorithms. Finally, a comprehensive database provides pedagogic value, as representative graphs from any considered sequence can be rapidly visualized.

To this end, we have created the *Encyclopedia of Finite Graphs* (henceforth Encyclopedia) which includes a database of graphs and invariants [16]. The software to fully populate the database has been released under an open source license [17]. The intention is for new invariants to be added to the project as various algorithms become available.

* Corresponding author. Tel.: +1 301 496 2618; fax: +1 301 496 0825.
E-mail address: hoppeta@mail.nih.gov (T. Hoppe).

http://dx.doi.org/10.1016/j.dam.2015.07.017
0166-218X/Published by Elsevier B.V.

Once built, the Encyclopedia readily yields integer sequences formed by matching the number of graphs to an invariant constraint at each order. Many such sequences have already been found and cataloged in another database, the Online Encyclopedia of Integer Sequences (OEIS) [35]. The OEIS was created by Neil Sloane in 1964 as a graduate student during his studies of combinatorial problems. Since then, the database has grown to over 250,000 sequences and is highly cited, with over 3,000 citations to date. The sequences are of general interest, spanning topics such as number theory, combinatorics, and graph theory. A given sequence may contain any number of terms, ranging from at least four up to as many as 500,000 (in the cases where the sequence admits a readily computable expression). Collecting and storing integer sequences in one place allows a researcher, who perhaps comes across the first few values of an unknown sequence, to be able to quickly look up subsequent values. The OEIS not only provides the numerical values but seeks to function as a true encyclopedia, with cross references to related sequences, references to other literature, and formulas when known. One of the primary goals of this paper is to systematically expand the sequences involving graph invariants known to the OEIS database. Through our exhaustive enumeration of small graphs, we were able to submit 141 new sequences to the OEIS and extend six existing sequences.

A graph invariant is any property that is preserved under isomorphism. Invariants can be simple binary properties (planarity), integers (automorphism group size), polynomials (chromatic polynomials), rationals (fractional chromatic numbers), complex numbers (adjacency spectra), sets (dominion sets) or even graphs themselves (subgraph and minor matching). We are primarily concerned with the sequences produced by graph invariants, i.e. the combinatorial problem of how many graphs of a given class satisfy a particular criteria. Let a graph be defined as the pair $G = (V, E)$, where V is a set of vertices and E is a set of edges. Define $\mathcal{C}$ as a class which forms an isomorphically distinct set of graphs that satisfy a specified criteria. Group the graphs into non-overlapping subsets such that

$$\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \dots \tag{1}$$

where $\mathcal{C}_n$ contains only graphs of order n. From here, define a sequence of subsets of the graph class

$$\mathcal{Q}(f, \mathcal{C}) = f(\mathcal{C}_1), f(\mathcal{C}_2), f(\mathcal{C}_3), \dots \tag{2}$$

where f is some invariant condition that selects from each set $\mathcal{C}_n$. Since $\mathcal{Q}$ selects from the graphs, we call $\mathcal{Q}$ the *query*.

Let $S(f, \mathcal{C}) = |f(\mathcal{C}_1)|, |f(\mathcal{C}_2)|, |f(\mathcal{C}_3)|, \dots$ be the sequence of integers defined by $\mathcal{Q}(f, \mathcal{C})$. For example, if $T(g)$ is the $\{0, 1\}$ indicator function that determines if the graph is a tree, and $\mathcal{C}'$ is the set of all simple unlabeled connected graphs, then

$$S(T(g) = 1, \mathcal{C}') = 1, 1, 1, 1, 2, 3, 6, 11, 23, \dots \tag{3}$$

which is sequence A000055 in the OEIS. This particular sequence is well-known and easily computable. We have evaluated a range of invariants, from those that are computable in polynomial time, to some that are known to be $\#\mathcal{P}$-complete. While a few sequences were merely extended, other invariants, such as the independence number, were hitherto unknown to OEIS and have produced novel sequences (A243781 –A243784).

In addition to contributing to the OEIS, a secondary goal is to identify relationships between graph invariants. Two queries of the same class are subsets of each other $\mathcal{Q}(f, \mathcal{C}) \subseteq \mathcal{Q}(g, \mathcal{C})$, if $f(\mathcal{C}_i) \subseteq g(\mathcal{C}_i)$ for all $i \geq 0$. Equality of two queries $\mathcal{Q}(f, \mathcal{C}) = \mathcal{Q}(g, \mathcal{C})$, implies $\mathcal{Q}(f, \mathcal{C}) \subseteq \mathcal{Q}(g, \mathcal{C})$ and $\mathcal{Q}(g, \mathcal{C}) \subseteq \mathcal{Q}(f, \mathcal{C})$. We say that a relation between two invariant conditions is *suggestive* to order n, $\mathcal{Q}(f, \mathcal{C}) \subseteq_n \mathcal{Q}(g, \mathcal{C})$, if $f(\mathcal{C}_i) \subseteq g(\mathcal{C}_i)$ for $0 \leq i \leq n$. They are *exclusive* to order n, $\mathcal{Q}(f, \mathcal{C}) \cap_n \mathcal{Q}(g, \mathcal{C})$, if $f(\mathcal{C}_i) \cap g(\mathcal{C}_i) = \emptyset$ for $0 \leq i \leq n$. We can quickly filter candidate relations by noting that it is necessary but not sufficient for the same conditions to hold for sequences, e.g. $S_a \neq S_b \Longrightarrow \mathcal{Q}(f, \mathcal{C}) \neq \mathcal{Q}(g, \mathcal{C})$.

We consider a final type of integer sequence, the number of distinct values an invariant could obtain for a given order. These sequences are not restricted to integer invariants. As an example, consider the integer sequence defined by the number of unique chromatic polynomials of graphs of a given order.

In this paper, we restrict the classes examined to those of simple connected graphs. Unless otherwise stated, any referenced graph is assumed to be simple and connected. Provided one had a means of enumeration, an extension of this program to other classes would be straightforward. Exhaustive generation algorithms are known for many specialized classes such as bipartite graphs, digraphs, multigraphs, regular graphs, cubic graphs, snarks, trees and maximal triangle-free graphs [24,25,7,6,33,8].

2. Methods

Using the `geng -c` command from `nauty` [24], we enumerated the simple connected graphs up to order $n \leq 10$. Our calculations drew upon a large number of open source libraries and tools. Many of the graph invariant calculations were done with either `networkx` [15] or `graph-tool` [10]. The invariants that were computable with integer or linear programming were done with PuLP [28]. For each graph we computed a series of invariants, which for completeness are described in Appendix A. A full table of all sequences submitted to the OEIS can be found in Appendix B. The relations between the invariants are numerous and are indexed online [17].

Since the graphs we consider are loopless and undirected, the edge incidence information for each graph requires $n(n-1)/2$ bits of storage. The largest we computed is of order 10, which requires 45 bits. This can be efficiently stored as a 64 bit unsigned integer using a binary representation. Graphs of order 11 would also be possible to be stored in this